

Proposal for Launch Vehicle for the NASA DRA 5.0

Dewey J., Galmiche T., Mejia M.

I. Nomenclature

ζ	= Correction factor
n_i	= Molecular composition of species "i"
A	= Area (m^2)
d	= Diameter (m)
r	= Radius (m)
I_{sp}	= Specific impulse (s)
\overline{m}	= Molecular weight (kg/mol)
C_F	= Thrust coefficient
C^*	= Characteristic velocity (m/s)
\dot{m}	= Mass flow rate (kg/s)
γ	= Ratio of specific heats
P	= Pressure (Pa)
M	= Mach number
T	= Temperature (k)
O/F	= Oxidizer to fuel ratio
ρ	= Density (kg/m^3)
C_d	= Discharge coefficient
N_{part}	= Number of certain parts
w	= Width (mm)
u	= Velocity of flow (m/s)
r_{c-sec}	= Radius of curvature of section (m)
L^*	= Characteristic length (m)
l_c	= Combustion chamber length (m)
λ	= Correction factor
θ_i	= Flow angle relative to axis of symmetry of the i^{th} point (deg)
$m_{0(i)}$	= Initial mass of ith stage (kg)
m_b	= Final mass of spacecraft after maneuver (kg)
Δv	= Change in velocity increment (m/s)
g_0	= Gravitational constant of 9.81 (m/s^2)

m_L	= Mass of payload (kg)
R	= Mass ratio
ϵ	= Structural ratio
N	= Number of stages
α	= Lagrange multiplier
λ_i	= Payload ratio of stage "i"
W	= Weight (N)
$t_{B(i)}$	= Burn time of ith stage (s)
Δt	= Change in time (s)
a_T	= Acceleration of the vehicle (m/s^2)
$\Psi_{t-\Delta t}$	= Flight path angle ($^\circ$)
alt_t	= Altitude (m)
$xdis_t$	= Downrange Distance (m)
θ	= Pitch angle ($^\circ$)

II. Introduction

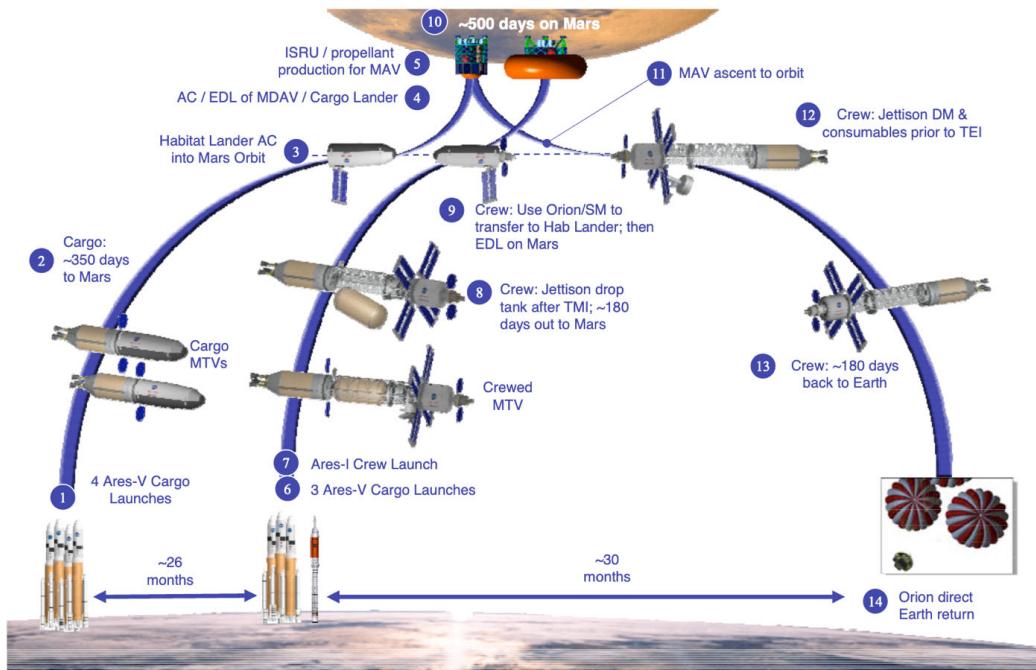


Figure 2-2. Mars Design Reference Architecture 5.0 mission sequence summary (NTR reference).

NASA's Design Reference Architecture 5.0 for a crewed mission to Mars calls for a heavy launch vehicle, the "Ares V" that will send multiple payloads into Low Earth Orbit. The first series of launches deliver cargo vehicles into orbit that then transit under their own power to Mars. These vehicles include supplies, a Mars ascent vehicle (MAV), a lander, and a

propellant production plant. Twenty-six months later when the orbits of Mars and Earth line up again for a Hohmann transfer, a series of Ares V vehicles launch three additional payloads that comprise the building blocks for Mars Transfer Vehicle (MTV) for the crew. The MTV is assembled from these components in orbit. The astronauts then launch on a human-rated vehicle, transfer to the MTV, and start a nine-month journey to Mars. As a pathfinding exercise for this architecture, NASA has released a Request for Proposals for designs of the “Ares V” launch vehicle. NASA requests that proposers present a detailed technical report on their vehicle design. The key requirements for the vehicle include

- A minimum of 120,000 kg delivered to an altitude of 400 km
- A minimum $\Delta v = 9.3 \text{ km/s}$
- Outer diameter greater than 10 m to accommodate the fairing
- A burnout velocity of 7.6 km/s, which is sufficient to enter a stable LEO at 400 km

In order to create this launch vehicle, we will begin by designing the engine for the first stage, the JMT-1, and the engine for the upper stages, JMT-2. From there, we will begin a trade study that will compare an architecture 1 with only two stages and an architecture 2 with three stages. Finally, we simulated the rocket launch and plotted its trajectory.

III. Engine 1 (JMT-1) Design

I. Key assumptions for analysis

We outline the following key assumptions that we will use in our evolution of this first rocket engine.

A. General engine characteristics

For this engine, we will be using LOX/RP-1 since it is widely available, of high density, and most importantly, liquid at room temperature. Table 1 provides initial estimations for combustion pressure, expansion ratio, and engine diameter from Rocketdyne’s F-1 engine as a baseline. These values will then be refined using CEA analysis. In order to reduce pressure losses in the combustion chamber, we will also assume that the combustion chamber area is 2 times the throat area.

Table 1: Initial Design Parameters

Propellant	Combustion Pressure	Expansion Ratio	Engine Diameter	Throat Diameter
LOX/RP-1	6.65 MPa	16	3.7 m	0.72 m [1]

B. Idealized assumptions

We employ the following idealized assumptions for our 2D thrust chamber analysis.

1. Nozzle

- Gas is homogeneously mixed
- Combustion occurs with negligible drift speed
- Nozzle is adiabatic downstream of combustion chamber

- No shocks inside combustion chamber
- Calorically perfect gas after combustion
- The engine is at sea level

2. Injector

- Single Pintle type
- Stiff injection pressure criteria
- Oxygen orifices are spaced 1 diameter apart from each other

3. Combustion chamber and throat region

- The radius of curvature leading to the throat from the combustion chamber is 125% of the throat radius. This curvature extends for an arc length of 45 degrees into the combustion chamber
 - Values are based on nozzle theory examples from Sutton textbook
- After 45 degrees, the combustion chamber radius is assumed to expand linearly up to the chamber diameter
- The overall length of the combustion chamber (from injector plate to throat) is determined by taking the shortest acceptable length from the L* criteria for LOX/RP-1

4. Nozzle design

- The radius of curvature downstream of the throat is 40% of the throat radius (up to inflection point)
- Shortened bell nozzle used
- This contour is determined from the Rao's method
- Final performance metrics at 75% of the way up from frozen flow estimates to shifting equilibrium estimates

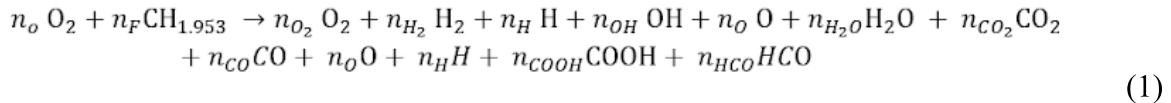
C. Empirical correction factors

For performance estimates, we use the following empirical correction factors for thrust and specific impulse. These account for a number of non-ideal effects such as flow divergence, boundary layer effects, unsteady combustion, nozzle erosion, and many others. The factors include:

- A thrust correction factor of $\zeta_F = 0.95$, which is the ratio of real to ideal thrust
- A discharge correction of $\zeta_d = 1.1$, which is the ratio of actual mass flow rate to the theoretical value

D. LOX/RP-1 reaction

LOX/RP-1 (a form of kerosene) is a more complex reaction giving rise to more combustion products. These include H₂O, CO₂, CO, H, O, O₃, HO₂, COOH, and HCO, which are governed by the reaction equation:



where n_i denotes the molar concentration of the i^{th} species. The use of $CH_{1.953}$ is an approximation for the significantly complex chemical formula for RP-1. This effectively represents the fact that on average, the chemical composition of RP-1 has 1.953 mols of hydrogen per mol of carbon. This is a useful truncation for calculating the resulting mixture content. With that said, as the reaction in Eq. 1 is a substantially more complex process, we will use the Chemical Equilibrium with Applications (CEA) tool for our analysis of this engine. CEA and the chemical properties of each species is handled internally by the solver, and as such, we do not list the propellant properties here.

II. Engine analysis

The following analysis inputs the initial design parameters into CEA to design a more efficient rocket engine.

A. Rocket performance

We consider the performance of the rocket in two limiting cases: frozen and shifting equilibrium flow—where we use the results from the previous sections as initial conditions for the flow’s expansion through the nozzle. We include plots of the change in Mach number, ratio of specific heats, temperature, pressure, density, and molecular weight as functions of the cross-sectional area of the rocket. We also estimate the performance metrics at the exit plane including thrust, specific impulse, and mass flow rate.

For frozen flow analysis, we must state a few assumptions that we are making for our calculations:

- The flow is adiabatic
- The flow is isentropic
- The molar ratios of products remains the same throughout the nozzle
- The specific heats can change with temperature

For shifting equilibrium analysis, we must again state a few assumptions that we are making for our calculations:

- The flow is isentropic
- The flow mixture is assumed to instantaneously equilibrate at each location in the nozzle.

Using the design parameters from the Rocketdyne F-1 engine shown in Table 1, we import these initial assumptions (shown in Table 1) into CEA to help us start designing our own fuel mixture and engine. From these initial parameters, we first calculated the throat area from the throat diameter using Eq. 2:

$$A_t = \pi \left(\frac{1}{2} d_t \right)^2 = \pi \left(\frac{1}{2} 0.72m \right)^2 = 0.4072 m^2 \quad (2)$$

We also proceeded to increase the internal combustion pressure of the engine to 10 MPa from the initial 6.65 MPa. This was done to try and get closer to the combustion pressure of SpaceX's Merlin-1D+ engine, and create a slightly more powerful version of the F-1 engine.

For the CEA analysis, we start by inputting the following options and values into the prompt windows (making slight alterations for the frozen flow analysis compared to the shifting equilibrium analysis):

Table 2: CEA Inputs

<u>Window:</u>	<u>Input Name:</u>	<u>Input Value/Option:</u> <u>Shifting Equilibrium/Frozen Flow</u>
Problem Type	Type Code	rocket
Pressures	Pressure	100 bar
Fuel(s)	Fuel Compound	RP-1
Oxidizer(s)	Oxidizer Compound	O2(L)
Oxid/Fuel	Specify Oxidizer/Fuel Ratio	o/f: Oxidizer/Fuel Wt. ratio
Oxid/Fuel	Low-Value	2
Oxid/Fuel	High-Value	9
Oxid/Fuel	Interval	0.5
Exit Condition	Supersonic Area Ratios	16
Final	Rocket Problem Options	Equilibrium/Frozen
Final	Rocket Problem Options	Infinite Area Combustor
Final	Post 'Submit' Options	Tabulate results
Transfer CEA Results into a Spreadsheet	Parameters	gam, mw, t, p, isp/ispfz, mach/machfz, cf/cffz

In these options shown above, we input a pressure of 100 bar, an O/F ratio range of 2 to 9 with an interval of 0.5, and an expansion ratio of 16 from Rocketdyne's F-1 engine. We then output the ratio of specific heats (γ, gam), the molecular weight (\bar{m}, mw), the adiabatic flame temperature (T_0, t), the combustion chamber pressure (P_0, p), the specific impulse ($I_{sp}, \text{isp} (\text{ispfz})$), the Mach number ($M, \text{mach} (\text{machfz})$), and the coefficient of thrust ($C_f, \text{cf} (\text{cffz})$).

Aero 535

Jorns

Fall 2022

With all of these values now tabulated into excel sheets from CEA, we can now plot the performance metrics through Matlab. To do this we start with these two initial equations:

$$Thrust = c^* c_F m \quad (3)$$

$$c^* = \frac{P_0 A_t}{m_i} \quad (4)$$

Both of these equations can be combined together, and along with the addition of the thrust correction factor lead to Eq. 5:

$$Thrust = \zeta_f P_0 A_t C_f \quad (5)$$

This pressure is plotted against all the different O/F ratios, and the combustion temperature and specific impulses are also plotted as their output from CEA as functions of O/F ratios. This results in Figure 1:

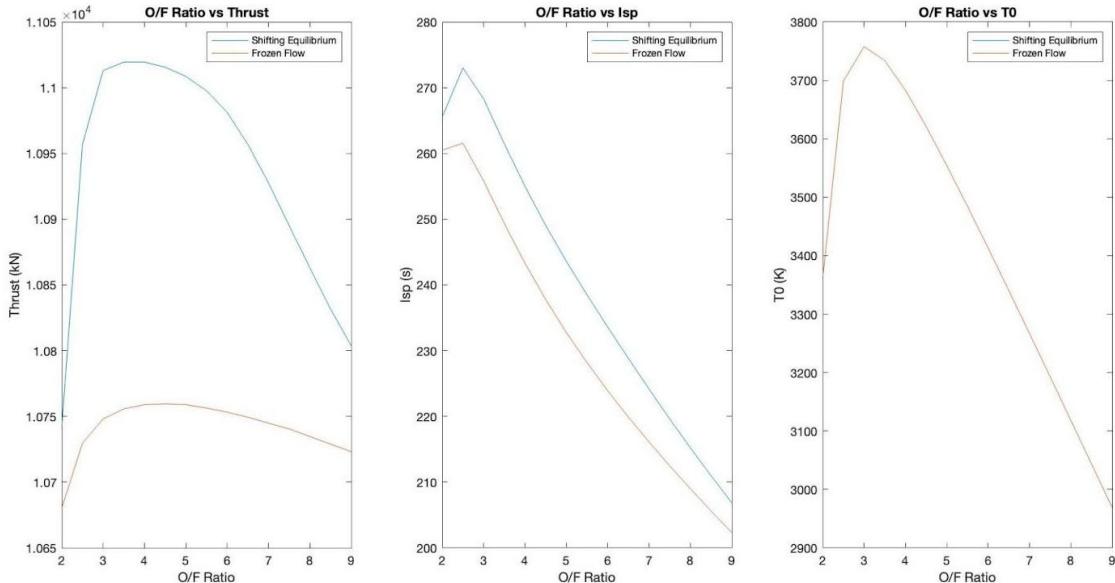


Figure 1: LOX/RP-1 Performance at Varying O/F Ratios

Plotting both frozen flow and shifting equilibrium, we see a peak in thrust at 3.5 for shifting and 4.5 for frozen, a peak in Isp at 2.5 for both frozen and shifting, and a peak in combustion temperature at 3 for both. Analyzing the dropoff in performance in each of these, we settled on an O/F ratio of 3 as it was near the peak of thrust while also not sacrificing too much Isp.

With a now settled O/F ratio of 3 we changed the inputs to CEA to now have an O/F ratio of only 3 and started varying the expansion ratio. This was done to help us further back out actual values analytically from our initial F-1 assumptions. For this study, we varied the expansion ratio between 10 and 150, taking measurements at every increment of 2 (slightly increasing this to 5 once we reached 50). We then once again recalculated thrust and plotted it

Aero 535

Jorns

Fall 2022

against the varying expansion ratio. Isp, temperature ratio (exit temperature vs. combustion temperature), molecular weight, pressure ratio (exit pressure vs. combustion pressure), and specific heat ratio were all extracted from CEA and plotted against these expansion ratio to analyze how the performance was affected by the varying expansion ratio.

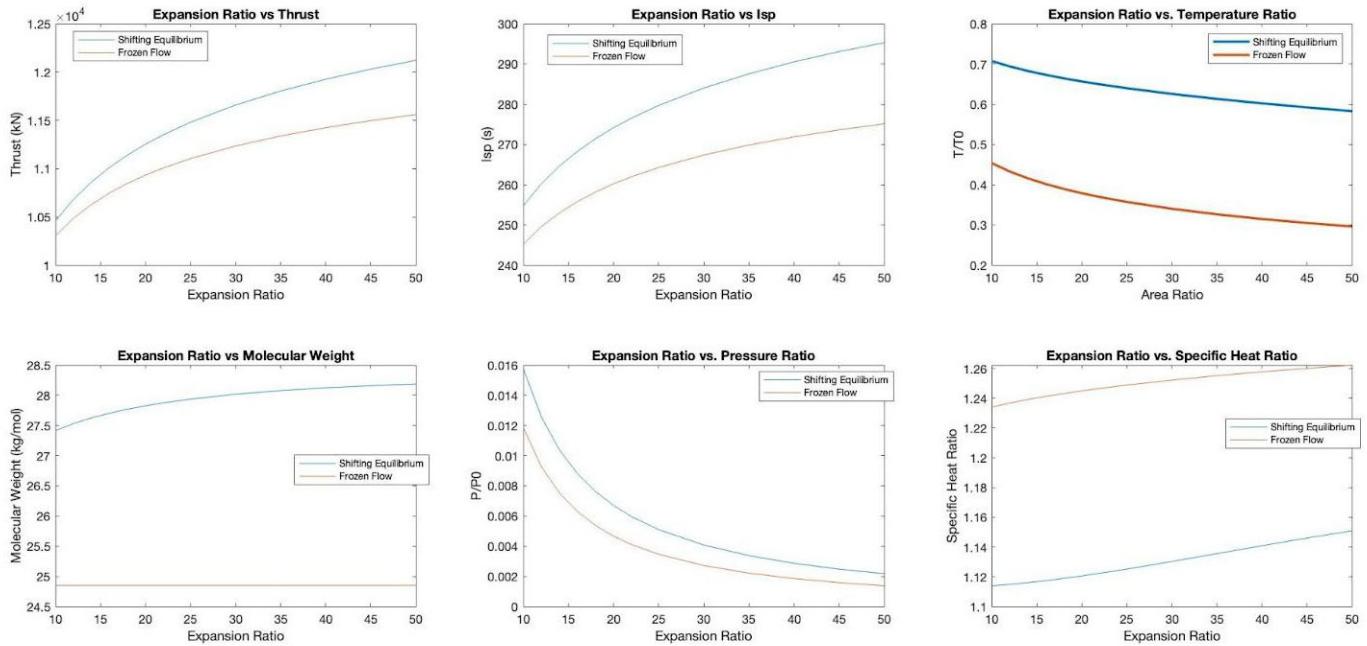


Figure 2: LOX/RP-1 Performance at Varying Expansion Ratios

From these plots, we decided to choose an expansion ratio of 30. This was chosen as it was about the area where most of these performance metrics started to flatten out and the benefits of increasing the expansion ratio became less and less worth it against the increase in size they would cause on the engine. Therefore to not unnecessarily create a massive engine, we settled with 30 as our expansion ratio. From here, we decided to keep the same combustion pressure and throat radius as we had been using thus far and decided to move forward with our design. With these finalized designed parameters, we re-ran another round of CEA analysis to extract the performance metrics before designing the nozzle:

Table 3: Engine Characteristics for JMT-1

Combustion Pressure	Expansion Ratio	Throat Area	Oxidizer - Fuel Ratio (mass)	Mean specific heat ratio(γ)	Mean molecular Weight	Combustion Temperature
10 MPa	30	0.407 m^2	3	<u>S.E.:</u> 1.13 <u>F.F.:</u> 1.25	28.01 g/mol 24.85 g/mol	3757.9 K

Engine	Propellant	Sea level thrust (kN)	Vacuum thrust (kN)	SL Isp (s)	Vacuum Isp (s)	Engine diameter	Mass Flow Rate (kg/s)
JMT-1	LOX/RP-1	<u>S.E.</u> : 7,064.9 <u>F.F.</u> : 6,807.8	<u>S.E.</u> : 7,538.5 <u>F.F.</u> : 7,123.2	<u>S.E.</u> : 284s <u>F.F.</u> : 267s	<u>S.E.</u> : 303s <u>F.F.</u> : 279s	3.7 m	<u>S.E.</u> : 2535.5 kg/s <u>F.F.</u> : 2595.2 kg/s

III. Nozzle analysis

A. Injector analysis

The purpose of an injector is to inject fuel and oxidizer at a certain design ratio, atomize, and then mix to achieve high combustion efficiency. For this engine and propellant, we will be using a pintle injector since it only requires one main element located in the center of the plate and is compatible with RP-1. The principle of operation is to have the oxidizer re-directed into a fan by a series of orifices arrayed radially around a central post. Fuel is then injected axially through this fan.

To begin the analysis, we calculate here the injector specifications for the oxidizer and fuel. The first step is to establish some parameters. From the previously mentioned assumptions, we can assign the combustion pressure to be $P_0 = 10 \text{ MPa}$. As for the oxidizer and fuel densities, lookup tables state that $\rho_{ox} = 1141 \text{ kg/m}^3$ and $\rho_f = 850 \text{ kg/m}^3$, respectively. The assumption of having a stiff injection pressure criteria implies that we need a minimum pressure drop that would prevent the propellant pumps from working too hard. The rule of thumb for injectors that are stiff to combustion instabilities is to use $\Delta P = 0.2 P_0 = 2 \text{ MPa}$.

After a few iterations, it was found that a fuel annulus width of 6.5 mm and an oxidizer orifice diameter (d_{ox}) of 3 mm and a corresponding discharge coefficient of 0.61 (C_d) was found to be the most efficient in ensuring that the pintle fit within the combustion chamber.

We can then determine the mass flow rate of the oxidizer at the injector. Assuming that the propellant is a fluid with subsonic speed, we can use the given equation

$$\dot{m}_{inj(ox)} = C_d A_{ox} \sqrt{2\rho_{ox} \Delta P} \quad (6)$$

where A_{ox} is the area of the oxidizer orifice.

To find A_{ox} , we can substitute the given value of d_{ox} into the equation for the area of a circle to find

$$A_{ox} = \pi \left(\frac{d_{ox}}{2} \right)^2 = 7.069 \text{ mm}^2 \quad (7)$$

Once we plug in the appropriate values back into Eq.6, we find that

$$\dot{m}_{inj(ox)} = 0.2913 \text{ kg/s} \quad (8)$$

The following step is to determine the total mass flow rate (\dot{m}). Assuming that the flow goes supersonic after the throat, we can implement choked flow relations in the throat where the Mach number (M) is assumed to be 1 to find

$$\dot{m} = \frac{\sqrt{\gamma} P_0 A_t M}{\sqrt{T_0 \frac{\bar{R}}{\bar{m}}}} \left[\left(\frac{1}{1 + \frac{\gamma-1}{2} M^2} \right)^{\frac{\gamma+1}{2(\gamma-1)}} \right] \quad (9)$$

where P_0 is the combustion chamber pressure, γ is the ratio of specific heats, A_t is the area of the throat, T_0 is the temperature of the combustion chamber, \bar{R} is the gas constant of 287 J/kgK, and \bar{m} is the mean molecular weight.

Plugging in values of $P_0 = 10 \text{ MPa}$, $\gamma = 1.13$, $A_t = 0.407 \text{ m}^2$, $T_0 = 3758 \text{ K}$ and

$\bar{m} = 28 \text{ g/mol}$ from Table 1 into Eq.9 gives us

$$\dot{m} = 2518 \text{ kg/s} \quad (10)$$

From here, we can calculate the mass flow rate of the oxidizer using the following given equation

$$\dot{m}_{ox} = \frac{\dot{m}}{1 + (F/O)} \quad (11)$$

where O/F is the oxidizer-fuel ratio.

By substituting the value of $O/F = 3$ from Table 1 and Eq.10 into Eq.11, we get

$$\dot{m}_{ox} = 1888.5 \text{ kg/s} \quad (12)$$

To find the mass flow rate of the fuel, we can simply subtract the mass flow rate of the oxidizer from the total mass flow rate

$$\dot{m}_f = \dot{m} - \dot{m}_{ox} = 629.5 \text{ kg/s} \quad (13)$$

We are then able to find the number of oxidizer injectors by calculating the ratio of the mass flow rate of the oxidizer to the mass flow rate of the injector

$$N_{inj(ox)} = \frac{\dot{m}_{ox}}{\dot{m}_{inj(ox)}} = 6483.1 \text{ holes} \quad (14)$$

This number is reported in Table 4.

The next step is to determine the pintle diameter. To do this, we must first find the area of the fuel orifice (A_f). Since we know the mass flow rate of the fuel, we can back out this value using a similar version of Eq. 6 but with the appropriate \dot{m} and density of fuel

$$\dot{m}_f = 76.368 = C_d A_f \sqrt{2\rho_f \Delta P} \rightarrow A_f = 0.0018 \text{ m}^2 \quad (15)$$

The assumptions state that the fuel orifice forms a 6.5 mm wide annulus surrounding the pintle which means that we can extract the pintle diameter from the overall area – the area of the fuel annulus. We must first subtract the area that includes the pintle radius and annulus from the area that includes just the pintle radius

$$A_f = \pi \left(\frac{D_p}{2} + w \right)^2 - \pi \left(\frac{D_p}{2} \right)^2 \quad (16)$$

where w denotes the width of the annulus and D_p is the pintle diameter.

Once we substitute Eq.15 and set $w = 0.0065 \text{ m}$ into Eq.16, we find that

$$D_p = 0.86 \text{ m} \quad (17)$$

This value is reported in Table 4.

Moving on, we now need to determine the number of rows of oxidizer orifices. But before that, we must first calculate how many orifices will be in each row. We can do this with the following equation with our given $d_{ox} = 0.001 \text{ m}$ and calculated Eq.17

$$N_{orf(ox)} = \frac{D_p \pi}{d_{ox}} = 900.8 \quad (18)$$

Since we assume that the oxygen orifices are spaced 1 diameter apart from each other, we need to divide the number of orifices per row by 2 to adjust for this

$$N_{orf(ox)} = \frac{1443}{2} = 450.4 \quad (19)$$

Finally, to determine the number of rows needed we can divide the number of oxidizer injectors from Eq.14 by our value in Eq.19 to get

$$\frac{N_{inj(ox)}}{N_{orf(ox)}} = 15 \text{ rows} \quad (20)$$

This value is reported in Table 4.

Table 4: Injector Specifications

Number of oxidizer injectors:	6484
Pintle diameter:	0.86 m
Number of rows of oxidizer orifices	15

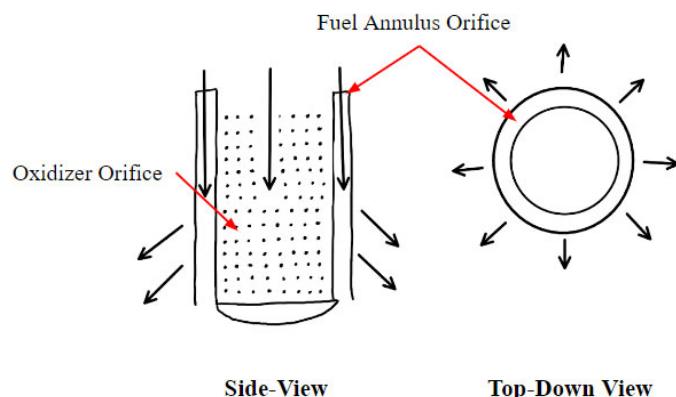


Figure 3: Cross-sectional sketch of pintle injector design

B. Propellant Delivery System

For a pump fed delivery on a launch vehicle, a powered turbopump increases the pressure of the propellant compressed gas from the propellant tank to the thrust chamber. By doing this there are much lower structural requirements and can deliver high chamber pressures. For this engine, we will be following the heritage of the F-1 engine and use a gas generator. This system is categorized as relatively simple with low turbine flow rates. The high temperatures of the gas generator products are able to support high pressure increases across the pumps and therefore higher chamber pressures as well. The pressure in the turbines is also relatively low since it can exhaust to ambient.

To determine how much power is required to run each pump, we need to use energy conservation across the pumps

$$\dot{m}_{pump} \left(\frac{u_{out}^2}{2} + \frac{P_{out}}{\rho_{out}} \right) - \dot{m}_{pump} \left(\frac{u_{in}^2}{2} + \frac{P_{in}}{\rho_{in}} \right) = Power_{pump} \quad (21)$$

where \dot{m}_{pump} is the mass flow rate through the pump, $u_{out \text{ or } in}$ is the flow velocity of the outlet and inlet respectively, $P_{out \text{ or } in}$ is the pressure of the outlet and inlet respectively, and $\rho_{out \text{ or } in}$ is the fluid density of the outlet and inlet respectively.

If we assume that the liquid is incompressible

$$\rho_{out} = \rho_{in} = \rho \quad (22)$$

and that kinetic energy is negligible at the inlet and outlet

$$u_{out}^2 \& u_{in}^2 \approx 0, \quad (23)$$

Then we can simplify the former equation to be

$$Power_{pump} = \frac{\dot{m}_{pump}(P_{out} - P_{in})}{\rho} \quad (24)$$

For \dot{m}_{pump} , we can use the previously calculated values of $\dot{m}_{ox} = 1889 \text{ kg/s}$ and $\dot{m}_f = 630 \text{ kg/s}$ from engine analysis as inputs for each pump. For ρ , we can use the previously stated lookup values of $\rho_{ox} = 1141 \text{ kg/m}^3$ and $\rho_f = 850 \text{ kg/m}^3$ as inputs.

For P_{out} , we want this value to be greater than our combustion chamber pressure, $P_0 = 10 \text{ MPa}$, to avoid backflow. Therefore we will use the rule of thumb and assume $P_{out} = 1.2P_0 = 12 \text{ MPa}$. Since we know that the rule of thumb for the inlet pressure is that P_{in} ranges from 0.07 and 0.3 MPa and that $P_{out} \gg P_{in}$, we will assume a value of 0.07 MPa. We can then plug in these respective values in Eq.24 to find that

$$\begin{aligned} Power_{fuel\ pump} &= 8.84\ MW \\ Power_{ox\ pump} &= 19.75\ MW \end{aligned} \quad (25)$$

These values are fairly reasonable as most cases have the oxidizer value be higher than the fuel one.

C. Combustion chamber contour

Here we calculate the contour of the nozzle from the injector plate up to the throat.

From the assumption that the radius of the curvature leading to the throat from the combustion chamber (r_{c-cham}) is 125% of the throat radius (r_t), we can use the following equation

$$r_{c-cham} = 1.25r_t \quad (26)$$

Since we are given a throat area of $A_t = 0.407\ m^2$ from Table 1, we can extract a throat radius from the equation for circle area

$$A_t = 0.407 = \pi r_t^2 \rightarrow r_t = 0.36\ m \quad (27)$$

We are then able to substitute Eq.27 into Eq.26 to find

$$r_{c-cham} = 0.45\ m \quad (28)$$

An important criterion for chamber design is that it is long enough so that combustion efficiency is high, around a value of 98%. In order to calculate the overall length of the combustion chamber from the injector to the throat (l_c), we need to use the equation for the characteristic chamber length (L^*)

$$L^* = \left(\frac{A_c}{A_t} \right) l_c \quad (29)$$

where A_c is the area of the combustion chamber.

We can assume that the combustion chamber area is 2 times the throat area in order to minimize stagnation pressure losses, which makes the contraction ratio in Eq.29 equal to 2. This number was chosen as an assumption because it was able to provide plots of reasonable contour whereas a value of 4 times the throat area did not. For our analysis, we then take the

shortest acceptable length from the L^* criteria for LOX/RP-1. Lookup tables state that this value is 102 cm. Substituting these numbers into Eq.29 gives us

$$1.02 \text{ m} = (2)l_c \rightarrow l_c = 0.51 \text{ m} \quad (30)$$

We assume that this curvature extends for an arc length of 45 degrees (θ) into the combustion chamber. This will form the first plot of the contour. The first step is to find the start and endpoints of the curve. The x-range will begin using the following equation and end at l_c , with increments of 0.0001 to ensure that several points are plotted smoothly

$$x_{start} = l_c - r_{c-cham} \sin(\theta) = 0.192 \text{ m} \quad (31)$$

$$x_{range1} = x_{start} : 0.0001 : l_c \quad (32)$$

where x_{range1} denotes the x-range at section 1.

With this range, we can now determine an equation to plot the values in the y-axis using a form of the circle equation:

$$(x - x_2)^2 + (y - y_2)^2 = r^2 \quad (33)$$

where (x_2, y_2) represents the origin and r is the radius of the circle, which in our case is the radius of the curvature from Eq.28.

Since we know that the first point is at the end of the combustion chamber and coincident with the throat, we can assume that the origin for Eq.33 is $(l_c, r_t + r_{c-cham})$. We then replace (x_2, y_2) by this origin to find that

$$y_{range1} = (r_t + r_{c-cham}) - \sqrt{r_{c-cham}^2 - (x - l_c)^2} \quad (34)$$

where y_{range1} denotes the y-range of section 1.

We can then plot our x-range from Eq.32 and y-range from Eq.34 to get the first section of the combustion chamber.

After 45 degrees, the combustion chamber radius is assumed to expand linearly up to the chamber diameter. This appears as a straight, slanted line from the curvature that ends once the endpoint reaches the radius of the chamber. In order to plot this second section of the chamber, we need to determine the start and end points of the line. We know that the starting point will be where the first section ended, at $[x_{range1}(1), y_{range1}(1)]$ where (1) represents the beginning of the curved portion of the chamber, as this is the intersection between the linearly expanding and curved sections.

As the radius expands linearly, we then know that the y-axis endpoint for this section will be at r_{cham} , the radius of the chamber. This can be found with the following equation

$$A_c = 2 * A_t = \pi r_{cham}^2 \rightarrow r_{cham} = 0.509 \text{ m} \quad (35)$$

For the x-axis endpoint, we use geometry of a right triangle at 45 degrees to determine

$$x_{end2} = x_{range1}(1) - \frac{r_{cham} - y_{range1}(1)}{\tan(\theta)} \quad (36)$$

where x_{end2} denotes the x-axis endpoint of section 2.

We can then plot our x-range from $[x_{end2}, x_{range1}(1)]$ and y-range from $[r_{cham}, y_{range1}(1)]$ to get the second section of the combustion chamber.

From there, the wall of the chamber becomes a straight section that is parallel to the engine centerline. Once again, the starting point will be where the second section ended, at $[x_{end2}, r_{cham}]$. As for the end point, the y-axis value will remain unchanged since it is a straight, flat line and the x-axis value will become 0 since it is where the combustion chamber begins. We can then plot our x-range from $[0, x_{end2}]$ and y-range from $[r_{cham}, r_{cham}]$ to get the third and final section of the combustion chamber. These three plots were combined using MATLAB and are shown in Figure 4 and 9.

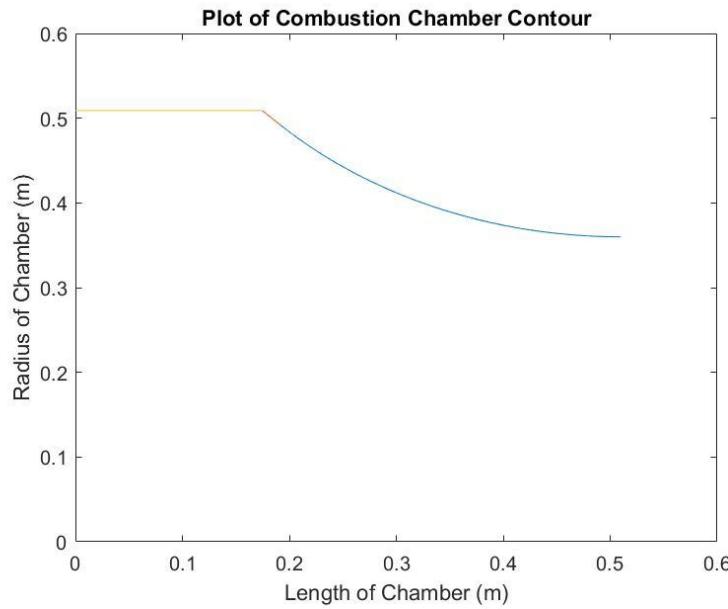


Figure 4: Plot of the combustion chamber contour for JMT-1

D. Bell nozzle

In designing the nozzle for our engines, we decided to use a shortened bell nozzle as this is the most common design for nozzles used in the industry. This is due to the fact that ideal nozzles that can provide isentropic expansion with minimal divergence losses are much too long, and short conical nozzles that are more easily manufactured produce higher divergences losses. Therefore, the shortened bell nozzle provides a comfortable middle-ground that allows the engine to keep much of the favorable characteristics of the ideal nozzle while also being a more manageable size.

We consider here the method derived by Rao for thrust optimized parabolic nozzles. The first step in this is to calculate the radius of curvature of the throat in the bell nozzle (r_{c-bell}) from Eq. 37:

$$r_{c-bell} = 0.4r_t \quad (37)$$

Plugging the now known throat radius from Table 1 into Eq. 38 we get:

$$r_c = 0.4 * 0.36m = 0.1440m \quad (38)$$

Knowing our initial performance metrics from the CEA analysis, we now move on to choosing a normalized length for the nozzle. To do this, we calculated the corrected performance in thrust at different normalized lengths using Eq. 39:

$$T = \lambda T_{initial} \quad (39)$$

Here λ represents the nozzle correction factor that affects the initially calculated thrust $T_{initial}$, which was initially found to be within the bounds of the shifting equilibrium and frozen flow analyses. These bounds are inputted as different thrusts into this equation. The nozzle correction factor is determined from Figure 5, where the intersection of the expansion ratio and the normalized length percentage gives us the correction factor.

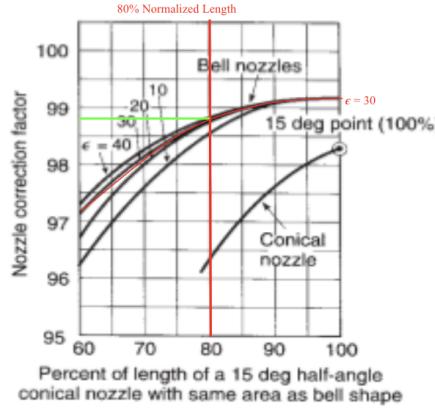


Figure 5: Nozzle Correction Factor vs. Normalized Length and Expansion Ratio

Not having chosen a specific normalized length yet, we extract the nozzle correction factor for normalized lengths of 60%, 70%, 80%, and 90%, and then plug the various correction factors back into Eq.39 to plot. This results in the following plot:

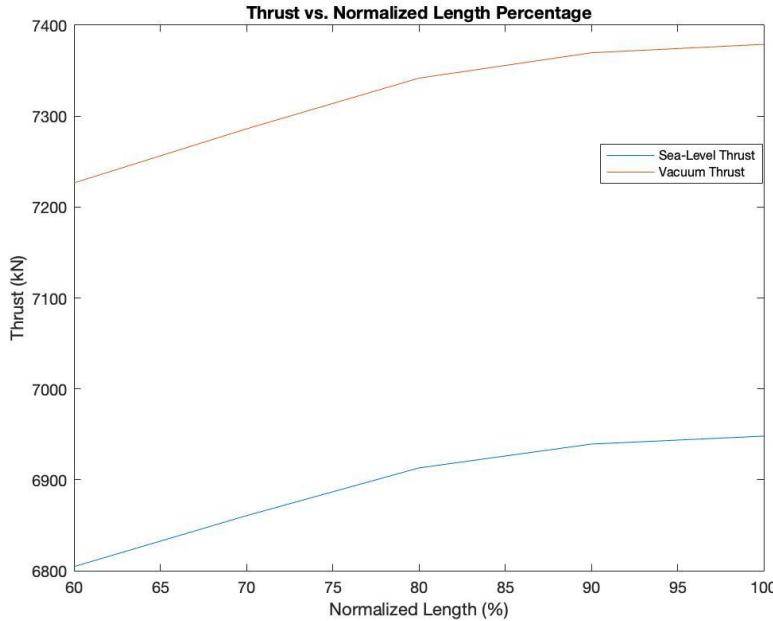


Figure 6: Thrust vs. Normalized Length

We found that after 80% the increase in thrust started to taper off, and therefore the added nozzle length was no longer worth the added weight and size. Knowing that the eventual length of the engine will not allow for the full chemical composition change to occur, we know that performance will be below the upper bound set by the shifting equilibrium analysis. Frozen flow sets the lower bound as it assumes no change occurs and therefore

hinders the values of performance. From past work and comparing past analysis to actual performances, we know that performance typically lies much closer to the shifting equilibrium analysis. Therefore, we decided to set our initial performance metrics at 75% of the way up from the frozen flow analysis to the shifting equilibrium analysis. This results in the following performance for our JMT-1 engine:

Table 5: Initial Engine Performance for JMT-1

Combustion Pressure	Expansion Ratio	Throat Area	Oxidizer - Fuel Ratio (mass)	Mean specific heat ratio(γ)	Mean molecular Weight	Combustion Temperature
10 MPa	30	0.407 m ²	3	1.13	28.01 g/mol	3757.9 K

Engine	Propellant	Sea level thrust (kN)	Vacuum thrust (kN)	SL Isp (s)	Vacuum Isp (s)	Engine diameter	Mass Flow Rate (kg/s)
JMT-1	LOX/RP-1	6,913 kN	7,341.7 kN	279.9 s	297.3 s	3.95 m	2,517.8

With our now decided normalized length of 80%, we were also able to extract final and initial parabola angles needed to plot the contour of the bell nozzle using Figure 7:

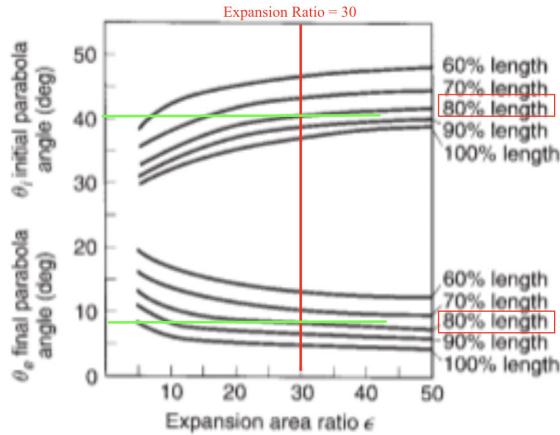


Figure 7: Parabola Angles vs. Expansion Ratio and Normalized Length

Applying our known expansion ratio of 30 and the normalized length of 80% we find angles:

$$\theta_i = 40.5^\circ, \theta_f = 8^\circ \quad (40)$$

The next step is to find the range of the expansion region of the nozzle. We first find the coordinates of the beginning of the expansion region of the nozzle using the following equations:

$$x = r_{c\text{-bell}} \sin\theta \quad y = r_t + r_{c\text{-bell}}(1 - \cos\theta) \quad (41)$$

We know the radius of curvature to be 0.1440 m from Eq.38, the throat radius was assumed to be 0.36 m in Table 1, and the initial inflection angle was just found to be 40.5°. Putting all this into the two equations from Eq.41, we get:

$$x = 0.0935m \quad y = 0.3945m \quad (42)$$

These two points give us the initial coordinate of the expansion region of the nozzle with the origin being at the axis of symmetry of the nozzle at the throat. Next, we find the end bound of the expansion region by calculating the radius of the exit plane using the following equation:

$$r_e = r_t \sqrt{\frac{A_e}{A_t}} = 1.9718 \text{ m} \quad (43)$$

The next step is to figure out the length of the nozzle from Eq.44. To do this we first find the length of a conical nozzle with a 15 degree half angle that has the same expansion ratio as our nozzle (L_{15}). This length can be found using the following equation:

$$L_{15} = \frac{r_t \left(\sqrt{\frac{A_e}{A_t}} - 1 \right) + r_c \left(\frac{1}{\cos(15^\circ)} - 1 \right)}{\tan(15^\circ)} \quad (44)$$

Plugging in our known values for the throat and chamber radius and the expansion ratio from Table 3, we find the calculated length is:

$$L_{15} = 6.0343 \text{ m} \quad (45)$$

Since here we are designing a shortened bell nozzle, we must now multiply this length by 0.8 to give us the normalized length of our nozzle:

$$L = 4.8274 \text{ m} \quad (46)$$

With our length and sea-level thrust now defined (from the Eq. 46 and Table 5 respectively), we can divide the two and find our thrust-to-length ratio (T/L) using Eq.47:

$$T/L = 6,913 \text{ kN} / 4.8274 \text{ m} = 1,432.0 \text{ kN/m} \quad (47)$$

The next step is to start plotting the contour of the nozzle, which is done by first solving a system of equations with the following 4 equations:

$$\begin{aligned} 0 &= Q + (T)^{1/2} \\ y'_e &= Px'_e + Q + (Sx'_e + T)^{1/2} \\ \tan\theta_i &= P + \frac{1}{2} \frac{S}{(T)^{1/2}} \\ \tan\theta_e &= P + \frac{1}{2} \frac{S}{(Sx'_e + T)^{1/2}} \end{aligned} \quad (48)$$

where P, Q, S, T are all unknown coefficients.

In this system, y'_e and x'_e can be easily calculated through the equations in Eq. 49:

$$\begin{aligned} x'_e &= L - r_c \sin\theta_i \\ y'_e &= r_e - r_t - r_c [1 - \cos\theta_i] \end{aligned} \quad (49)$$

After inputting the values from previous equations, we are able to find the values to be:

$$\begin{aligned} x'_e &= 4.7339 \\ y'_e &= 1.5773 \end{aligned} \quad (50)$$

Inputting these into the 4 equations from Eq. 48 and running that through the *solve* function of MATLAB we are able to find the coefficients to be:

$$\begin{aligned} P &= -0.2783 \\ Q &= -1.6989 \\ S &= 3.8474 \\ T &= 2.8862 \end{aligned} \quad (51)$$

This helps us find the overall equation for the contour of the parabola of our nozzle:

$$y' = Px' + Q + (Sx' + T)^{1/2} \quad (52)$$

To calculate the y-coordinates for the parabola, we assume that it starts at x=0 and then feed it data points of x-values from 0 to the previously calculated x'_e value that signifies the exit plane of our nozzle. Now having the x and y coordinates of the parabola, we offset the coordinates to start at the end of the curvature and plot both together to get Figure 8:

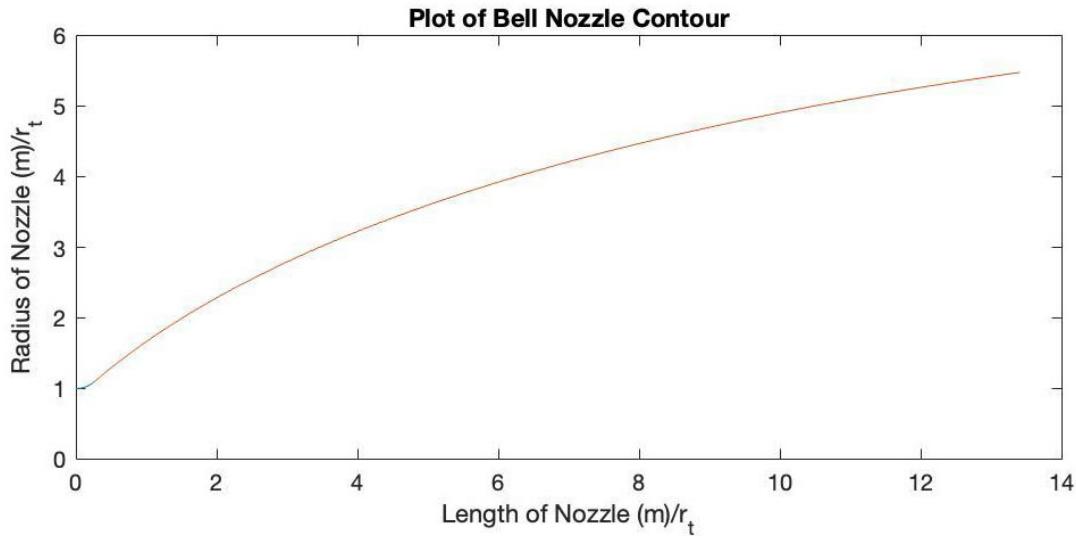


Figure 8: Bell Nozzle Contour

Putting this plot along with the plot of the combustion chamber from Figure 4 we get the full-length contour of the half-engine:

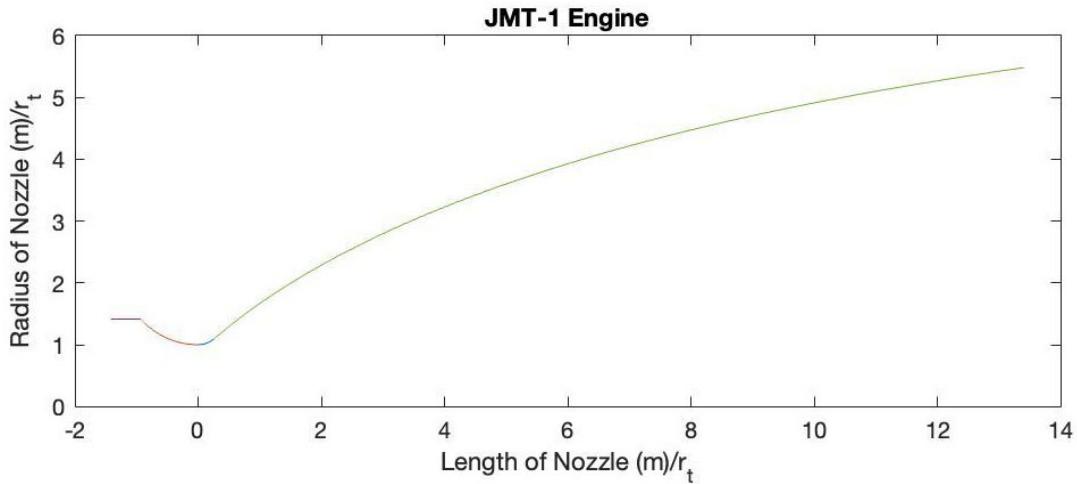


Figure 9: JMT-1 Engine Contour

IV. Thermal management description

For the thermal management of the thrust chamber, we need to take into account the heated gas from the combustion chamber that will transmit heat to the surrounding walls of the engine. This total power that is lost to the walls often occurs through radiation and convection of heat through the boundary layer. The biggest concern is the high power density that is coincident with the throat. The associated thermal flux can sometimes be high enough to melt the material of the walls in a matter of seconds and therefore needs an efficient strategy for thermal management.

We have chosen the steady-state method of regenerative cooling for the JMT-1 engine. Also used by Rocketdyne's F-1 engine, this strategy has the cool RP-1 fuel flow through channels around the periphery of the thrust chamber and absorb the heat as it passes through. Once the fuel has been heated, it is then injected into the combustion chamber.

IV. Engine 2 (JMT-2) Design

I. Key assumptions for analysis

We outline the following key assumptions that we will use in our evolution of this second rocket engine.

A. General engine characteristics

For this engine, we will be using LOX/LH₂ since it has the highest performance of any flown propellants and uses clean products like water vapor. Table 6 provides initial estimations for combustion pressure, expansion ratio, and throat area from the Saturn V rocket's J-2 engine as a baseline. These values will then be refined using CEA analysis. In order to reduce pressure losses in the combustion chamber, we will also assume that the combustion chamber area is 2.5 times the throat area.

Table 6: Design parameters for second engine

Propellant	Combustion Pressure	Expansion Ratio	Throat Area
LOX/LH ₂	5.3 MPa	27.5	0.1092 m ²

B. Idealized assumptions

We employ the following idealized assumptions for our 2D thrust chamber analysis.

1. Nozzle

- Gas is homogeneously mixed
- Combustion occurs with negligible drift speed
- Nozzle is adiabatic downstream of combustion chamber
- No shocks inside combustion chamber
- Calorically perfect gas after combustion
- The engine is at sea level

2. Injector

- Swirl Coaxial type
- Stiff injection pressure criteria
- Oxidizer orifice diameter of 6.5 mm
- Discharge coefficient of 0.61
- These values were chosen using Table 8-2 in the Sutton textbook based on "sharped edged" orifice type

3. Combustion chamber and throat region

- The radius of curvature leading to the throat from the combustion chamber is 125% of the throat radius. This curvature extends for an arc length of 45 degrees into the combustion chamber
 - Values are based on nozzle theory examples from Sutton textbook (102)
- After 45 degrees, the combustion chamber radius is assumed to expand linearly up to the chamber diameter
- The overall length of the combustion chamber (from injector plate to throat) is determined by taking the shortest acceptable length from the L* criteria for LOX/LH₂

4. Nozzle design

- The radius of curvature downstream of the throat is 40% of the throat radius (up to inflection point)
- Shortened bell nozzle used
- This contour is determined from the Rao's method
- Final performance metrics at 75% of the way up from frozen flow estimates to shifting equilibrium estimates

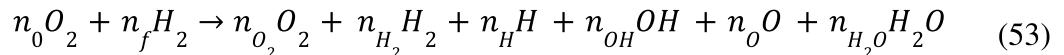
C. Empirical correction factors

For performance estimates, we use the following empirical correction factors for thrust and specific impulse. These account for a number of non-ideal effects such as flow divergence, boundary layer effects, unsteady combustion, nozzle erosion, and many others. The factors include:

- A thrust correction factor of $\zeta_F = 0.95$, which is the ratio of real to ideal thrust
- A discharge correction of $\zeta_d = 1.1$, which is the ratio of actual mass flow rate to the theoretical value

D. LOX/LH₂ reaction

For combustion of hydrogen (H₂) and oxygen (O₂), we assume the possible combustion products are H₂O, H, O, OH, O₂, H₂:



There are two other possible products, ozone (O₃) and peroxide (H₂O₂); however, these are highly unstable and quickly break down into the other products. On the time scales of interest, they thus can be ignored in the reaction. We will use the Chemical Equilibrium with Applications (CEA) tool for our analysis of this engine. CEA and the chemical properties of each species is handled internally by the solver, and as such, we do not list the propellant properties here.

II. Engine analysis

The following analysis inputs the initial design parameters into CEA to design a more efficient rocket engine.

A. Rocket performance

We consider the performance of the rocket in two limiting cases: frozen and shifting equilibrium flow—where we use the results from the previous sections as initial conditions for the flow’s expansion through the nozzle. We include plots of the change in Mach number, ratio of specific heats, temperature, pressure, density, and molecular weight as functions of the cross-sectional area of the rocket. We also estimate the performance metrics at the exit plane including thrust, specific impulse, and mass flow rate.

For frozen flow analysis, we must state a few assumptions that we are making for our calculations:

- The flow is adiabatic
- The flow is isentropic
- The molar ratios of products remains the same throughout the nozzle
- The specific heats can change with temperature

For shifting equilibrium flow analysis, we must again state a few assumptions that we are making for our calculations:

- The flow is isentropic
- The flow mixture is assumed to instantaneously equilibrate at each location in the nozzle.

Using the same process as was used for the JMT-1 engine, we pulled the initial values from the J-2 engine over to CEA for a shifting equilibrium and frozen flow analysis of a LOX/LH₂ mixture. We started with the same selection process by varying the O/F ratio:

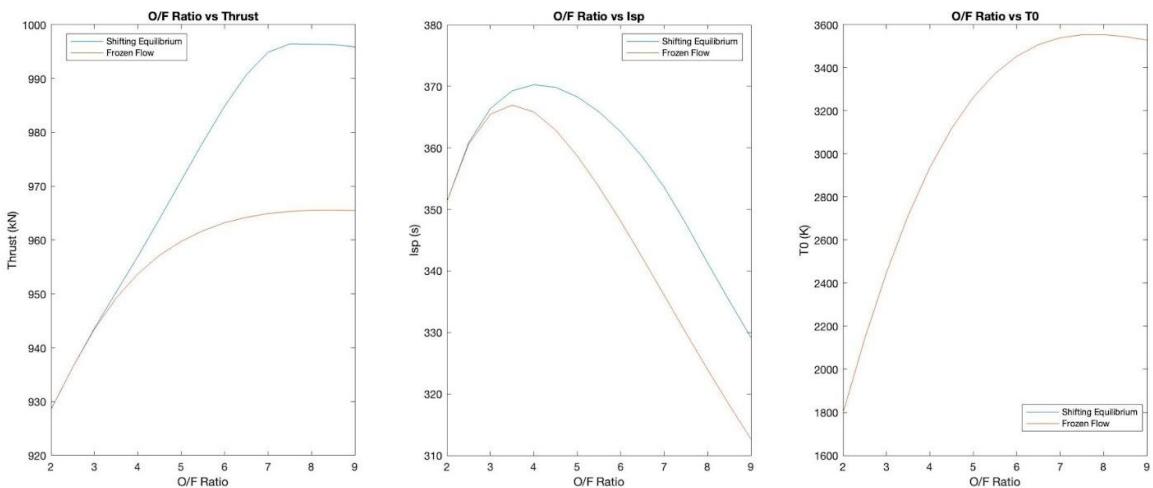


Figure 10: LOX/LH₂ Performance at Varying O/F Ratios

From the study, we saw a spike in thrust at an O/F ratio of 7.5, spikes in Isp at O/F ratios of 3.5 and 4.5, and a spike in combustion temperature at an O/F ratio of 8. As this engine will primarily be used in vacuum conditions to boost the upper stages, we decided to choose an O/F ratio of 4 to maximize the Isp performance as the thrust values will be higher than shown here as these values represent sea-level performance. This selection now allowed us to move onto the sizing of the expansion ratio as was done with the JMT-1 engine.

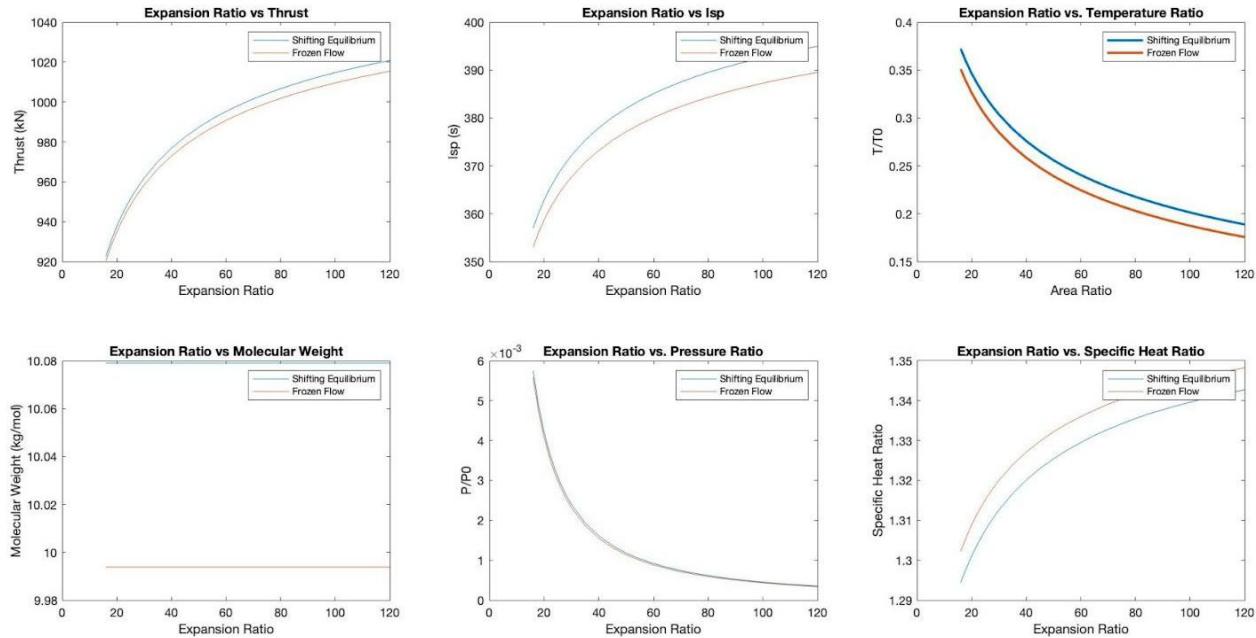


Figure 11: LOX/LH₂ Performance at Varying Expansion Ratios

From the performance metrics we decided to choose an expansion ratio of 60. This was chosen as it was about the area where most of these performance metrics started to flatten out and the benefits of increasing the expansion ratio became less and less worth it against the increase in size they would cause on the engine. Therefore to not unnecessarily create a massive engine, we settled with 30 as our expansion ratio. From here, we decided to keep the same combustion pressure and throat radius as we had been using thus far and decided to move forward with our design. With these finalized designed parameters, we re-ran another round of CEA analysis to extract the performance metrics before designing the nozzle:

Table 7: Engine characteristics for JMT-2

Combustion Pressure	Expansion Ratio	Throat Area	Oxidizer - Fuel Ratio (mass)	Mean specific heat ratio(γ)	Mean molecular Weight	Combustion Temperature
5.3 MPa	60	0.109 m ²	4	<u>S.E.:</u> 1.33 <u>F.E.:</u> 1.34	10.10 g/mol 9.99 g/mol	2940 K

Table 8: Performance Metric Boundaries for JMT-1 Engine

Engine	Propellant	Sea level thrust (kN)	Vacuum thrust (kN)	SL Isp (s)	Vacuum Isp (s)	Engine diameter	Mass Flow Rate (kg/s)
JMT-2	LOX/LH2	<u>S.E.:</u> 995.2 <u>F.F.:</u> 990.7	<u>S.E.:</u> 1,025.5 <u>F.F.:</u> 1,020.0	<u>S.E.:</u> 385s <u>F.F.:</u> 380s	<u>S.E.:</u> 396s <u>F.F.:</u> 391s	2.89 m	<u>S.E.:</u> 263.46 kg/s <u>F.F.:</u> 265.75 kg/s

III. Nozzle analysis

A. Injector analysis

For this engine and propellant, we will be using a coaxial post injector since it works well with cryogenic propellants where one has been gasified by regenerative heating. The principle of operation is that liquid oxidizer flows up a hollow post. The gaseous propellant flows in the sleeve surrounding the post. The concentric streams then emerge at substantially different velocities and densities. This promotes a shear-driven instability that facilitates rapid atomization and mixing through large-scale oscillations. The injector will have a multiplicity of these coaxial posts on its face.

To begin the analysis, we must first establish some parameters. From the previously mentioned assumptions, we can assign the oxygen orifice diameter to be $d_{ox} = 6.5 \text{ mm}$, the combustion pressure to be $P_0 = 5.3 \text{ MPa}$, and the discharge coefficient to be 0.61. As for the oxidizer and fuel densities, lookup tables state that $\rho_{ox} = 1141 \text{ kg/m}^3$ and $\rho_f = 70.85 \text{ kg/m}^3$, respectively. The assumption of having a stiff injection pressure criteria implies that we need a minimum pressure drop that would prevent the propellant pumps from working too hard. The rule of thumb for injectors that are stiff to combustion instabilities is to use $\Delta P = 0.2P_0 = 1.06 \text{ MPa}$.

We can then determine the mass flow rate of the oxidizer at the injector. Assuming that the propellant is a fluid with subsonic speed, we can use the given equation

$$\dot{m}_{inj(ox)} = C_d A_{ox} \sqrt{2\rho_{ox} \Delta P} \quad (54)$$

where A_{ox} is the area of the oxidizer orifice.

To find A_{ox} , we can substitute the given value of d_{ox} into the equation for the area of a circle to find

$$A_{ox} = \pi \left(\frac{d_{ox}}{2} \right)^2 = 33.18 \text{ mm}^2 \quad (55)$$

Once we plug in the appropriate values back into Eq. 54, we find that

$$\dot{m}_{inj(ox)} = 0.996 \text{ kg/s} \quad (56)$$

The following step is to determine the total mass flow rate (\dot{m}). Assuming that the flow goes supersonic after the throat, we can implement choked flow relations in the throat where the Mach number (M) is assumed to be 1 to find

$$\dot{m} = \frac{\sqrt{\gamma P_0 A_t M}}{\sqrt{T_0 \bar{R}}} \left[\left(\frac{1}{1 + \frac{\gamma-1}{2} M^2} \right)^{\frac{\gamma+1}{2(\gamma-1)}} \right] \quad (57)$$

where P_0 is the combustion chamber pressure, γ is the ratio of specific heats, A_t is the area of the throat, T_0 is the temperature of the combustion chamber, \bar{R} is the gas constant of 287 J/kgK, and \bar{m} is the mean molecular weight.

Plugging in values of $P_0 = 5.3 \text{ MPa}$, $\gamma = 1.33$, $A_t = 0.109 \text{ m}^2$, $T_0 = 2940 \text{ K}$ and $\bar{m} = 10.10 \text{ g/mol}$ from Table into Eq.57 gives us

$$\dot{m} = 260.7 \text{ kg/s} \quad (58)$$

From here, we can calculate the mass flow rate of the oxidizer using the following given equation

$$\dot{m}_{ox} = \frac{\dot{m}}{1 + (F/O)} \quad (59)$$

where O/F is the oxidizer-fuel ratio.

By substituting the value of $O/F = 4$ from Table 7 and Eq.58 into Eq.59, we get

$$\dot{m}_{ox} = 208.56 \text{ kg/s} \quad (60)$$

To find the mass flow rate of the fuel, we can simply subtract the mass flow rate of the oxidizer from the total mass flow rate

$$\dot{m}_f = \dot{m} - \dot{m}_{ox} = 52.14 \text{ kg/s} \quad (61)$$

We are then able to find the number of oxidizer injectors by calculating the ratio of the mass flow rate of the oxidizer to the mass flow rate of the injector

$$N_{inj(ox)} = \frac{\dot{m}_{ox}}{\dot{m}_{inj(ox)}} = 209.5 \quad (62)$$

This number is reported in Table 9.

The next step is to match the number of oxidizer injectors to the number of fuel injectors. To do this, we must first assume an initial value for the fuel annulus width (w). The geometry of a coaxial post injector indicates that the radius of the injector is simply

$$r_{inj} = \frac{d_{ox}}{2} + w \quad (63)$$

This equation can then be plugged into area of the fuel annulus

$$A_f = \pi(r_{inj} + w)^2 - \pi r_{inj}^2 \quad (64)$$

From here, we can determine the mass flow rate of the fuel orifice using a similar version of Eq.54

$$\dot{m}_{inj(f)} = C_d A_f \sqrt{2\rho_f \Delta P} \quad (65)$$

We are then able to see if the number of fuel injectors match that of the oxidizer orifice by adjusting the width of the fuel annulus

$$N_{inj(f)} = \frac{\dot{m}_f}{\dot{m}_{inj(f)}} \quad (66)$$

$$N_{inj(ox)} = N_{inj(f)} \quad (67)$$

After a few iterations, we found that a fuel annulus width of 1.1 mm was able to match the number of injectors. We were also able to determine the diameter of the entire coaxial injector itself with the following equation

$$D_{coax} = \frac{N_{inj}(r_{inj}/2)}{\pi} = 0.5802 \text{ m} \quad (68)$$

Table 9: Injector Specifications

Number of fuel and oxidizer injectors:	1309
Coaxial diameter:	0.58 m
Fuel annulus width	1.1 mm

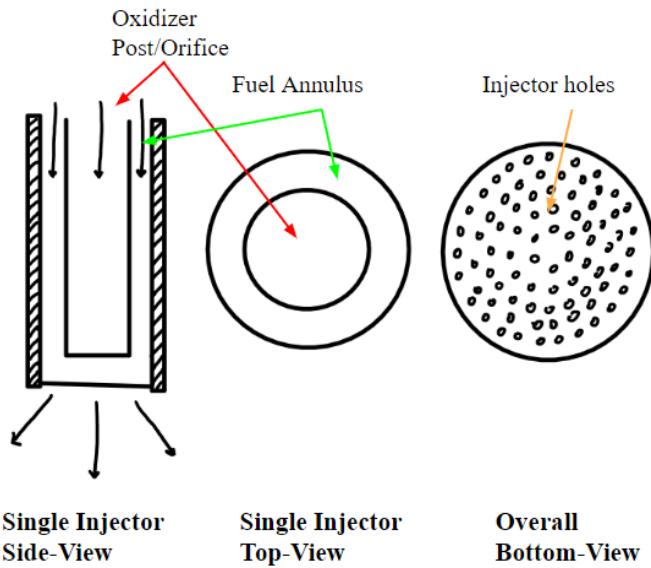


Figure 12: Cross-sectional sketch of coaxial injector design

B. Propellant Delivery System

For a pump fed delivery on a launch vehicle, a powered turbopump increases the pressure of the propellant compressed gas from the propellant tank to the thrust chamber. By doing this there are much lower structural requirements and can deliver high chamber pressures. For this engine, we will be following the heritage of the upper stage Atlas V and Delta IV and use an expander cycle type system. This type is known as a “closed cycle” where all propellant is sent through the combustion chamber, which yields a higher specific impulse.

The low temperature of the fluid at the turbines helps preserve its lifetime. Overall, the expander cycle is simple and provides low engine mass.

To determine how much power is required to run each pump, we will make the same assumptions of the fluid being incompressible and kinetic energy being negligible at the inlet and outlet and use the same simplified equation (mentioned in JMT-1 pump system analysis) where

$$\text{Power}_{\text{pump}} = \frac{\dot{m}_{\text{pump}}(P_{\text{out}} - P_{\text{in}})}{\rho} \quad (69)$$

For \dot{m}_{pump} , we can use the previously calculated values of $\dot{m}_{\text{ox}} = 208.56 \text{ kg/s}$ and $\dot{m}_f = 52.14 \text{ kg/s}$ from engine analysis as inputs for each pump. For ρ , we can use the previously stated lookup values of $\rho_{\text{ox}} = 1141 \text{ kg/m}^3$ and $\rho_f = 70.85 \text{ kg/m}^3$ as inputs. For P_{out} , we want this value to be greater than our combustion chamber pressure, $P_0 = 5.3 \text{ MPa}$, to avoid backflow. Therefore we will use the rule of thumb and assume $P_{\text{out}} = 1.2P_0 = 6.36 \text{ MPa}$. Since we know that the rule of thumb for the inlet pressure is that P_{in} ranges from 0.07 and 0.3 MPa and that $P_{\text{out}} \gg P_{\text{in}}$, we will assume a value of 0.07 MPa. We can then plug in these respective values in Eq.57 to find that

$$\text{Power}_{\text{fuel pump}} = 4.63 \text{ MW} \quad (70)$$

$$\text{Power}_{\text{ox pump}} = 1.15 \text{ MW}$$

Normally the oxidizer value is higher but we believe the fuel value is higher in this case due to the vast fluid density disparity between LOX and LH2.

C. Combustion chamber contour

Here we calculate the contour of the nozzle from the injector plate up to the throat.

From the assumption that the radius of the curvature leading to the throat from the combustion chamber (r_{c-cham}) is 125% of the throat radius (r_t), we can use the following equation

$$r_{c-cham} = 1.25r_t \quad (71)$$

Since we are given a throat area of $A_t = 0.109 \text{ m}^2$ from Table 6, we can extract a throat radius from the equation for circle area

$$A_t = 0.109 = \pi r_t^2 \rightarrow r_t = 0.186 \text{ m} \quad (72)$$

We are then able to substitute Eq.72 into Eq.71 to find

$$r_{c-cham} = 0.233 \text{ m} \quad (73)$$

An important criterion for chamber design is that it is long enough so that combustion efficiency is high, around a value of 98%. In order to calculate the overall length of the combustion chamber from the injector to the throat (l_c), we need to use the equation for the characteristic chamber length (L^*)

$$L^* = \left(\frac{A_c}{A_t} \right) l_c \quad (74)$$

where A_c is the area of the combustion chamber.

We can assume that the combustion chamber area is 2.5 times the throat area in order to minimize stagnation pressure losses, which makes the contraction ratio in Eq.73 equal to 2.5. For our analysis, we then take the shortest acceptable length from the L^* criteria for LOX/LH₂. Lookup tables state that this value is 76 cm. Substituting these numbers into Eq.74 gives us

$$0.76 \text{ m} = (2.5)l_c \rightarrow l_c = 0.304 \text{ m} \quad (75)$$

This value is reported in Table 4.

We assume that this curvature extends for an arc length of 45 degrees (θ) into the combustion chamber. This will form the first plot of the contour. The first step is to find the start and endpoints of the curve. The x-range will begin using the following equation and end at l_c , with increments of 0.0001 to ensure that several points are plotted smoothly

$$x_{start} = l_c - r_{c-cham} \sin(\theta) = 0.139 \text{ m} \quad (76)$$

$$x_{range1} = x_{start} : 0.0001 : l_c$$

where x_{range1} denotes the x-range at section 1.

With this range, we can now determine an equation to plot the values in the y-axis using a form of the circle equation:

$$(x - x_2)^2 + (y - y_2)^2 = r^2 \quad (77)$$

where (x_2, y_2) represents the origin and r is the radius of the circle, which in our case is the radius of the curvature from Eq.73.

Since we know that the first point is at the end of the combustion chamber and coincident with the throat, we can assume that the origin for Eq.23 is $(l_c, r_t + r_{c-cham})$. We then replace (x_2, y_2) by this origin to find that

$$y_{range1} = (r_t + r_{c-cham}) - \sqrt{r_{c-cham}^2 - (x - l_c)^2} \quad (78)$$

where y_{range1} denotes the y-range of section 1.

We can then plot our x-range from Eq.76 and y-range from Eq.78 to get the first section of the combustion chamber.

After 45 degrees, the combustion chamber radius is assumed to expand linearly up to the chamber diameter. This appears as a straight, slanted line from the curvature that ends once the endpoint reaches the radius of the chamber. In order to plot this second section of the chamber, we need to determine the start and end points of the line. We know that the starting point will be where the first section ended, at $[x_{range1}(1), y_{range1}(1)]$ where (1) represents the beginning of the curved portion of the chamber, as this is the intersection between the linearly expanding and curved sections.

As the radius expands linearly, we then know that the y-axis endpoint for this section will be at r_{cham} , the radius of the chamber. This can be found with the following equation

$$A_c = 2.5 * A_t = \pi r_{cham}^2 \rightarrow r_{cham} = 0.294 \text{ m} \quad (79)$$

For the x-axis endpoint, we use geometry of a right triangle at 45 degrees to determine

$$x_{end2} = x_{range1}(1) - \frac{r_{cham} - y_{range1}(1)}{\tan(\theta)} \quad (80)$$

where x_{end2} denotes the x-axis endpoint of section 2.

We can then plot our x-range from $[x_{end2}, x_{range1}(1)]$ and y-range from $[r_{cham}, y_{range1}(1)]$ to get the second section of the combustion chamber.

From there, the wall of the chamber becomes a straight section that is parallel to the engine centerline. Once again, the starting point will be where the second section ended, at $[x_{end2}, r_{cham}]$. As for the end point, the y-axis value will remain unchanged since it is a straight, flat line and the x-axis value will become 0 since it is where the combustion chamber begins. We can then plot our x-range from $[0, x_{end2}]$ and y-range from $[r_{cham}, r_{cham}]$ to get the third and final section of the combustion chamber. These three plots were combined using MATLAB and are shown in Figure 13 and 18.

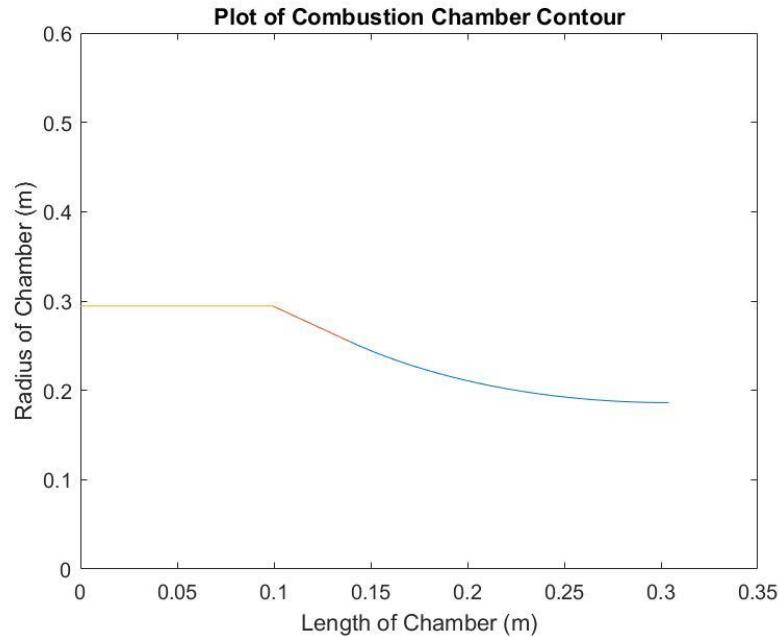


Figure 13: Plot of the combustion chamber contour for JMT-2

D. Bell nozzle

In designing the nozzle for our engines, we decided to use a shortened bell nozzle as this is the most common design for nozzles used in the industry. This is due to the fact that ideal nozzles that can provide isentropic expansion with minimal divergence losses are much too long, and short conical nozzles that are more easily manufactured produce higher divergence losses. Therefore, the shortened bell nozzle provides a comfortable middle-ground that

allows the engine to keep much of the favorable characteristics of the ideal nozzle while also being a more manageable size.

We consider here the method derived by Rao for thrust optimized parabolic nozzles. The first step in this is to calculate the radius of curvature of the throat in the bell nozzle (r_{c-bell}) from Eq. 81:

$$r_{c-bell} = 0.4r_t \quad (81)$$

Plugging the now known throat radius from Table 6 into Eq.82 we get:

$$r_c = 0.4 * 0.1092m = 0.0437m \quad (82)$$

Knowing our initial performance metrics from the CEA analysis, we now move on to choosing a normalized length for the nozzle. To do this, we calculated the corrected performance in thrust at different normalized lengths using Eq. 83:

$$T = \lambda T_{initial} \quad (83)$$

Here λ represents the nozzle correction factor that affects the initially calculated thrust $T_{initial}$, which was initially found to be within the bounds of the shifting equilibrium and frozen flow analyses. These bounds are inputted as different thrusts into this equation. The nozzle correction factor is determined from Figure 14, where the intersection of the expansion ratio and the normalized length percentage gives us the correction factor.

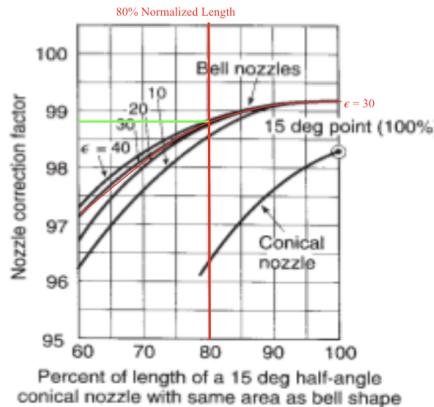


Figure 14: Nozzle Correction Factor vs. Normalized Length and Expansion Ratio
Not having chosen a specific normalized length yet, we extract the nozzle correction factor for normalized lengths of 60%, 70%, 80%, and 90%, and then plug the various correction factors back into Eq.83 to plot. This results in the following plot:

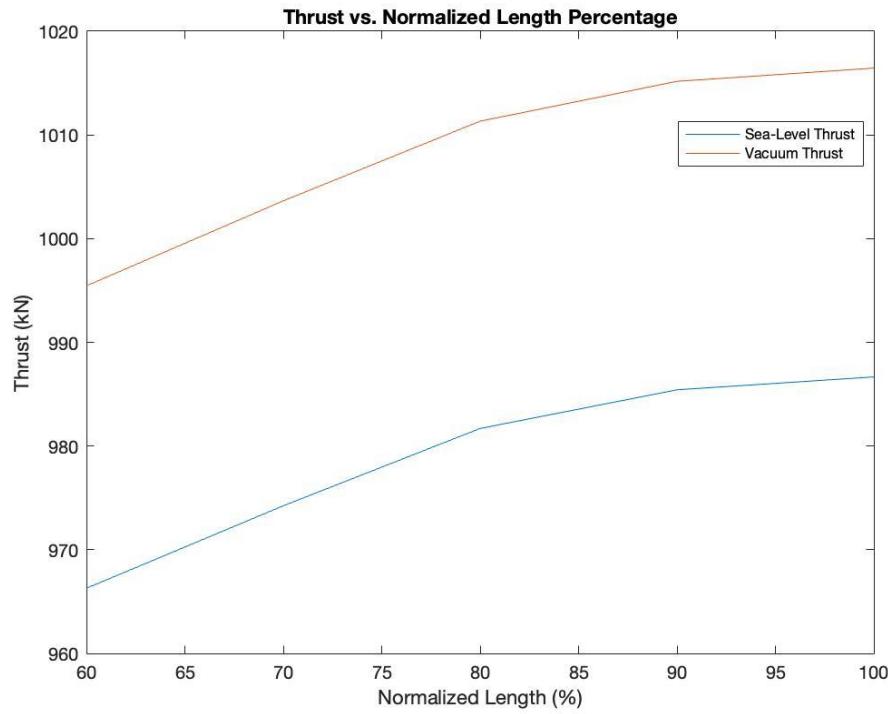


Figure 15: Thrust vs. Normalized Length

We found that after 80% the increase in thrust started to taper off, and therefore the added nozzle length was no longer worth the added weight and size. Knowing that the eventual length of the engine will not allow for the full chemical composition change to occur, we know that performance will be below the upper bound set by the shifting equilibrium analysis. Frozen flow sets the lower bound as it assumes no change occurs and therefore hinders the values of performance. From past work and comparing past analysis to actual performances, we know that performance typically lies much closer to the shifting equilibrium analysis. Therefore, we decided to set our initial performance metrics at 75% of the way up from the frozen flow analysis to the shifting equilibrium analysis. This results in the following performance for our JMT-2 engine:

Table 10: Initial Engine Performance for JMT-2

Combustion Pressure	Expansion Ratio	Throat Area	Oxidizer - Fuel Ratio (mass)	Mean specific heat ratio(γ)	Mean molecular Weight	Combustion Temperature
5.3 MPa	60	0.0375 m ²	4	1.19	10.08 g/mol	2937.4 K

Engine	Propellant	Sea level thrust (kN)	Vacuum thrust (kN)	SL Isp (s)	Vacuum Isp (s)	Engine diameter	Mass Flow Rate (kg/s)
--------	------------	-----------------------	--------------------	------------	----------------	-----------------	-----------------------

JMT-2	LOX/LH2	981.7 kN	1,011.3 kN	383.8 s	395.4s	1.70 m	260.7 kg/s
-------	---------	----------	------------	---------	--------	--------	------------

With our now decided normalized length of 80%, we were also able to extract final and initial parabola angles needed to plot the contour of the bell nozzle using Figure 16:

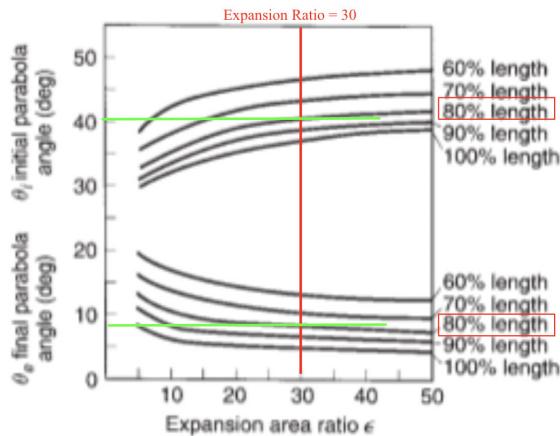


Figure 16: Parabola Angles vs. Expansion Ratio and Normalized Length

Applying our known expansion ratio of 30 and the normalized length of 80% we find angles:

$$\theta_i = 40.5^\circ, \theta_f = 8^\circ \quad (84)$$

The next step is to find the range of the expansion region of the nozzle. We first find the coordinates of the beginning of the expansion region of the nozzle using the following equations:

$$x = r_{c-bell} \sin \theta \quad y = r_t + r_{c-bell} (1 - \cos \theta) \quad (85)$$

We know the radius of curvature to be 0.1440 m from Eq.82, the throat radius was assumed to be 0.36 m in Table 7, and the initial inflection angle was just found to be 40.5°. Putting all this into the two equations from Eq.85, we get:

$$x = 0.0284m \quad y = 0.1197m \quad (86)$$

These two points give us the initial coordinate of the expansion region of the nozzle with the origin being at the axis of symmetry of the nozzle at the throat. Next, we find the end bound of the expansion region by calculating the radius of the exit plane using the following equation:

$$r_e = r_t \sqrt{\frac{A_e}{A_t}} = 0.8459 \text{ m} \quad (87)$$

The next step is to figure out the length of the nozzle from Eq.88. To do this we first find the length of a conical nozzle with a 15 degree half angle that has the same expansion ratio as our nozzle (L_{15}). This length can be found using the following equation:

$$L_{15} = \frac{r_t \left(\sqrt{\frac{A_e}{A_t}} - 1 \right) + r_c \left(\frac{1}{\cos(15^\circ)} - 1 \right)}{\tan(15^\circ)} \quad (88)$$

Plugging in our known values for the throat and chamber radius and the expansion ratio from Table 10, we find the calculated length is:

$$L_{15} = 2.7550 \text{ m} \quad (89)$$

Since here we are designing a shortened bell nozzle, we must now multiply this length by 0.8 to give us the normalized length of our nozzle:

$$L = 2.2040 \text{ m} \quad (90)$$

With our length and sea-level thrust now defined (from the Eq. 90 and Table 10 respectively), we can divide the two and find our thrust-to-length ratio (T/L) using Eq.91:

$$T/L = 981.7 \text{ kN} / 2.2040 \text{ m} = 445.41 \text{ kN/m} \quad (91)$$

The next step is to start plotting the contour of the nozzle, which is done by first solving a system of equations with the following 4 equations:

$$\begin{aligned} 0 &= Q + (T)^{1/2} \\ y'_e &= Px'_e + Q + (Sx'_e + T)^{1/2} \\ \tan\theta_i &= P + \frac{1}{2} \frac{S}{(T)^{1/2}} \\ \tan\theta_e &= P + \frac{1}{2} \frac{S}{(Sx'_e + T)^{1/2}} \end{aligned} \quad (92)$$

In this system, y'_e and x'_e can be easily calculated through the equations in Eq. 93:

$$\begin{aligned} x'_e &= L - r_c \sin\theta_i \\ y'_e &= r_e - r_t - r_c [1 - \cos\theta_i] \end{aligned} \quad (93)$$

After inputting the values from previous equations, we are able to find the values to be:

$$\begin{aligned}x_e' &= 2.1756 \\y_e' &= 0.7262\end{aligned}\tag{94}$$

Inputting these into the 4 equations from Eq. 92 and running that through the *solve* function of MATLAB we are able to find the coefficients to be:

$$\begin{aligned}P &= -0.2811 \\Q &= -0.7904 \\S &= 1.7944 \\T &= 0.6247\end{aligned}\tag{95}$$

This helps us find the overall equation for the contour of the parabola of our nozzle:

$$y' = Px' + Q + (Sx' + T)^{1/2}\tag{96}$$

To calculate the y-coordinates for the parabola, we assume that it starts at $x=0$ and then feed it data points of x-values from 0 to the previously calculated x_e' value that signifies the exit plane of our nozzle. Now having the x and y coordinates of the parabola, we offset the coordinates to start at the end of the curvature and plot both together to get Figure 17:

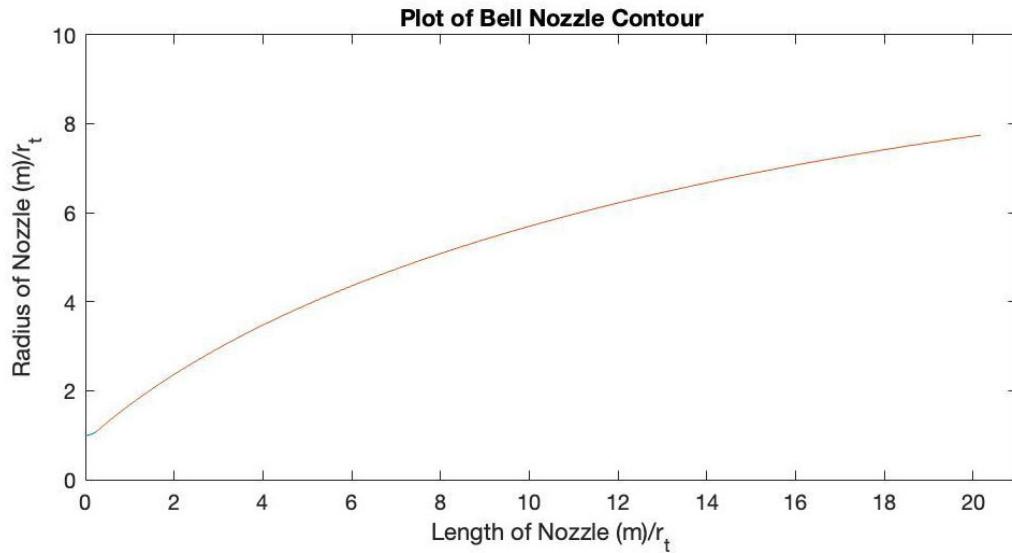


Figure 17: Bell Nozzle Contour

Putting this plot along with the plot of the combustion chamber from Figure 13 we get the full-length contour of the half-engine:

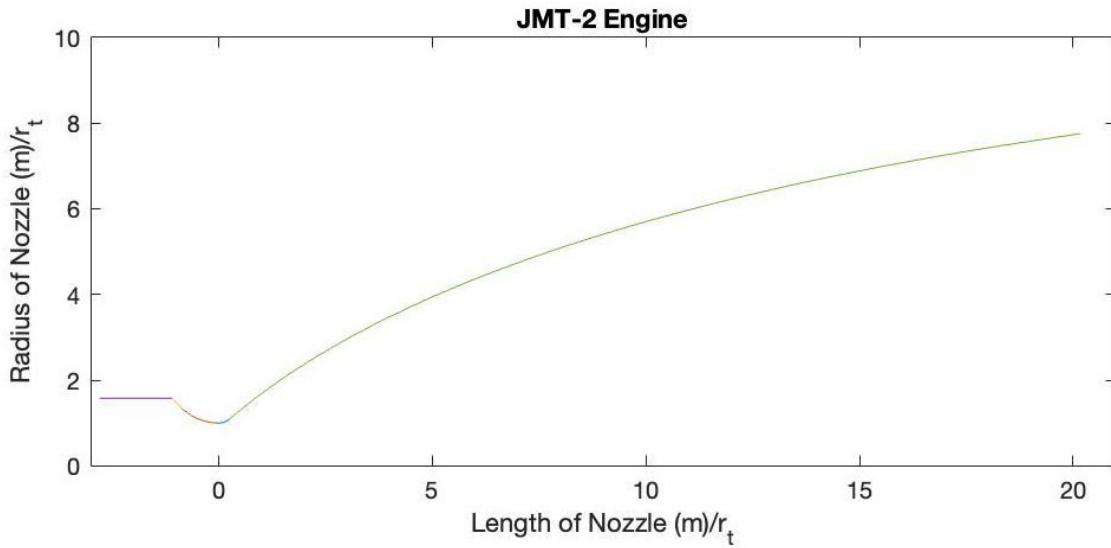


Figure 18: JMT-2 Engine Contour

IV. Thermal management description

In order to manage the thermal load of high power densities at the throat, we have chosen the steady-state method of regenerative cooling for the JMT-2 engine. Since the LH₂ fuel has to be cryogenically stored, it is already cool enough to flow through channels around the periphery of the thrust chamber and absorb the heat as it passes through. Once the fuel has been heated and emerges gasified, it is then injected into the combustion chamber using the coaxial post injector that is designed to work well with gasified propellants. By using this method, thermal energy avoids being wasted and is re-absorbed before combustion. This can even improve the average specific impulse by 1% which is useful for this engine that will be designed to be used in the upper stages of the launch vehicle after the first stage has left the atmosphere.

V. Launch vehicle configuration

For our launch vehicle trade study, we will be considering two architectures: a rocket with 2 stages and the other with 3 stages. The first stage for both will have LOX/RP-1 propellant and the upper stages will utilize LOX/LH₂.

I. Key assumptions for analysis

We outline in the following the key assumptions we use in our trade study. This includes a discussion of the payload mass, the delta-v budget, and the propulsion characteristics of each architecture.

A. Payload mass

Both launch vehicle architectures will have the same payload mass, which will consist of supplies, a MAV, a lander, and a propellant production plant. After several months the

rocket will launch three additional payloads that comprise the building blocks for Mars Transfer Vehicle (MTV) for the crew. We are given a minimum payload mass of 120,000 kg that needs to be delivered.

B. Delta-v budget

Since both architectures will have the same payload, they will also have the same delta-v requirements. The main maneuver of concern is leaving Earth's atmosphere and launching cargo to LEO. A minimum $\Delta v = 9.3 \text{ km/s}$ will be needed to transport everything to an altitude of 400 km. After initial calculations we found that a $\Delta v = 9500 \text{ km/s}$ would give us the more reasonable values, so that is the value we will be using for our sizing and trajectory. A burnout velocity of 7.6 km/s, which is sufficient to enter a stable LEO, will also be required for each architecture.

C. Average properties for propulsion system

We show in Table 11 and 12 the expected specific impulse for the different stages of the mission.

Table 11: Average properties of architecture 1

Propellant	Engine	Average specific impulse	Structural coefficient (ϵ)	Thrust to weight at ignition
LOX/RP-1	1st stage	289 s	0.05	1.5
LOX/LH2	2nd stage	395 s (vacuum)	0.09	0.8

Table 12: Average properties of architecture 2

Propellant	Engine	Average specific impulse	Structural coefficient (ϵ)	Thrust to weight at ignition
LOX/RP-1	1st stage	289 s	0.05	1.2
LOX/LH2	2nd stage	390 s (average vacuum + s.l.)	0.07	0.9
LOX/LH2	3rd stage	395 s (vacuum)	0.19	0.5

For the launch vehicle stages of each architecture, we chose to baseline a kerosene first stage (LOX/RP-1) and liquid hydrogen (LX/LH2) upper stages. RP-1 has a lower upper bound in achievable specific impulse compared to liquid hydrogen. This translates to a lower fuel economy, implying that more propellant is required for our delta-v. However, the LOX/RP-1 mixture has a density that is higher than the LOX/LH2 mixture. This indicates that a launch vehicle using LOX/RP-1 will require less volume for propellant storage than a vehicle with LOX/LH2. This is a critical consideration for the lower stage which will spend most of its time in the Earth's atmosphere. The increased volume of the

rocket becomes a prohibitively large source drag for a LOX/LH₂ first stage. But once the launch vehicle is above the atmosphere about to reach the required 400 km, upper stages are ignited and atmospheric drag is no longer a concern. The higher performance LOX/LH₂ is therefore the choice for the 2nd stages of both architectures and the 3rd stage of the second architecture.

The estimates for specific impulse in Table 11 and 12 represent average values expected for each stage based on the previous analysis on each engine. For the RP-1 first stage, 289 s was chosen as an average from sea level and vacuum specific impulse values, found in Table 13. The second stage of the second architecture was also chosen as an average between sea level and vacuum numbers. These stages will mostly be in the Earth's atmosphere. As for the second and third stages of architecture 1 and 2 respectively, 395 s was chosen from the vacuum condition extreme since these stages will be beyond the atmosphere

We include in Table 11 and 12 estimates for the structural coefficients for the launch vehicles. These represent the ratios of structural mass in each stage to the total wet mass of the stage. For architecture 1, we chose values of 0.05 and 0.09 for stage 1 and 2 respectively. These values were found from literature review that compared optimized 2-stage launch vehicles with the same propellant combinations [2]. Since RP-1 is a denser propellant, it requires less volume for storage. The storage tanks consequently are smaller allowing for lower structural coefficient whereas LH requires larger volume and more tank mass for its storage. Similarly, since the pressure differential in the tanks will grow higher with altitude, there is a need for more structural mass on the 3rd stage of the second architecture. The estimates for architecture 2 were based on our past mission analysis that consisted of the same number of stages and propellant.

Finally, we show in Table 11 and 12 the requirements for the thrust to weight of each vehicle stage. For the first stage, the initial trajectory is vertical as the vehicle climbs out of the Earth's atmosphere. Therefore sufficient acceleration needs to be provided. thrust to weight ratio greater than 1 allows for the rocket to overcome the force of gravity. Because of this, we used estimations of 1.5 and 1.2 for the architectures, with architecture being slightly higher due to the fact that there are only 2 stages in comparison to 3. In upper stages as the rocket trajectory gains altitude and shifts its trajectory to horizontal with respect to the surface of the earth, the need for acceleration decreases. This decrease allowed for us to estimate an equivalent decrease in the subsequent thrust to weight ratios.

D. Baseline engines for trajectory analysis

We show in Table 13 the properties of our two engines, the RP-1 based JMT-1 engine and the LH₂ based JMT-2 engine. While we use the average performance metrics in Table 11 and 12 to evaluate our initial design, we return to the properties shown in Table 13 to estimate the number of engines and total thrust levels required for each stage.

Table 13: Properties of the engines considered for the rocket stages.

Engine	Propellant	Sea level thrust (kN)	Vacuum thrust (kN)	SL Isp (s)	Vacuum Isp (s)	Mass of engine (kg)	Engine diameter
JMT-1	LOX/RP-1	6,913.0	7,341.7	279.9	297.3	8400	3.95 m
JMT-2	LOX/LH2	981.7	1,011.3	383.8	395.4	1788	1.70 m

II. Analysis

We present in the following our design and sizing for the launch vehicle.

A. Mass of each stage

The goal of this section is to use the minimum required payload mass combined with the delta-v requirement for the launch vehicle to determine the optimal masses of each stage for both architectures.

1. Architecture 1

Our first step here was to calculate the mass ratio (R_i) for each of the two stages using the equation:

$$R_i = \frac{M_{0(i)}}{M_{b(i)}} = \frac{\alpha I_{sp(i)} g_0 + 1}{\alpha I_{sp(i)} g_0 \epsilon_i} \quad (97)$$

where α denotes the Lagrange Multiplier, $I_{sp(i)}$ is the specific impulse for the “ith” stage, and ϵ_i is the structural coefficient for the “ith” stage.

We then plug Eq. 97 into the following equation for delta-v

$$\Delta v = 9.5 \text{ km/s} = \sum_{i=1}^2 I_{sp(i)} g_0 \ln(R_i) \quad (98)$$

After substituting in the appropriate $I_{sp(i)}$ and ϵ_i from Table 11 into Eq. 98, we are able to find the Lagrange Multiplier to be

$$\alpha = -0.0004276 \quad (99)$$

This allows us to plug Eq.99 back into Eq.97 to find exact numbers of R_i for each stage

$$R_1 = 3.5028, R_2 = 4.4055 \quad (100)$$

Now that we have values in Eq.100, we can substitute this into the payload ratio for each stage (λ_i)

$$\lambda_i = \frac{1 - \epsilon_i R_i}{R_i - 1}, \quad (101)$$

and find that

$$\lambda_1 = 0.3296, \lambda_2 = 0.1772 \quad (102)$$

To determine the total initial mass for each rocket, we use the optimal mass ratio:

$$\frac{m_{0(i)}}{m_L} = \prod_{i=1}^2 \left(\frac{\lambda_i + 1}{\lambda_i} \right) \quad (103)$$

where $m_{0(i)}$ denotes the initial mass for the “ith” stage and m_L is the payload mass.

We substitute Eq. 102 to find that

$$\frac{m_{0(i)}}{m_L} = 26.7993 \quad (104)$$

To calculate the total initial masses for each rocket, we plug in the value of m_L of 120,000 kg from the problem statement into Eq.104 and were able to find

$$m_{0(1)} = 3,215,921.92 \text{ kg} \quad (105)$$

Aero 535

Jorns

Fall 2022

These masses are recorded under “Total mass including payload” in Table 14.

The final step is to calculate the masses for each stage. To do this, we use another equation for payload ratio:

$$\lambda_i = \frac{m_{0(i+1)}}{m_{0(i)} - m_{0(i+1)}} \quad (106)$$

where $m_{0(i+1)}$ denotes the initial mass of the “i+1” stage.

Knowing the payload ratio for each stage and having the mass of the payload for each architecture, we were then able to work our way down the rocket using Eq. 106 to find the overall rocket mass at the ignition of each stage

$$\begin{aligned} m_{0(2)} &= m_L \left(\frac{1+\lambda_2}{\lambda_2} \right) = 797,149.49 \text{ kg} \\ m_{0(1)} &= m_{0(2)} \left(\frac{1+\lambda_1}{\lambda_1} \right) = 3,215,921.92 \text{ kg} \end{aligned} \quad (107)$$

We can then use these values to find the mass of each stage

$$\begin{aligned} m_{2nd} &= m_{0(2)} - m_L = 677,149.40 \text{ kg} \\ m_{1st} &= m_{0(1)} - m_{0(2)} = 2,418,772.43 \text{ kg} \end{aligned} \quad (108)$$

Finally, we can calculate the dry mass of each stage ($m_{b(i)}$) using the following equation

$$R_i = \frac{M_{0(i)}}{M_{b(i)}} \quad (109)$$

Plugging in the appropriate values for R_i and $M_{0(i)}$ of each stage we find that the dry / burn out mass is

$$m_{b(1st)} = 918,088.11 \text{ kg}, \quad m_{b(2nd)} = 180,943.45 \text{ kg} \quad (110)$$

2. Architecture 2

Our first step here was to calculate the mass ratio (R_i) for each of the three stages using the equation:

$$R_i = \frac{M_{0(i)}}{M_{b(i)}} = \frac{\alpha I_{sp(i)} g_0 + 1}{\alpha I_{sp(i)} g_0 \epsilon_i} \quad (111)$$

where $m_{b(i)}$ is the mass after the burn out, also known as the dry mass.

We then plug Eq. 111 into the following equation for delta-v

$$\Delta v = 9.5 \text{ km/s} = \sum_{i=1}^3 I_{sp(i)} g_0 \ln(R_i) \quad (112)$$

After substituting in the appropriate $I_{sp(i)}$ and ϵ_i from Table 12 into Eq. 112, we are able to find the Lagrange Multiplier to be

$$\alpha = -0.0003856 \quad (113)$$

This allows us to plug Eq.113 back into Eq.111 to find exact numbers of R_i for each stage

$$R_1 = 1.7047, R_2 = 4.6019, R_3 = 1.7406 \quad (114)$$

Now that we have values in Eq.114, we can substitute this into the payload ratio for each stage (λ_i)

$$\lambda_i = \frac{1 - \epsilon_i R_i}{R_i - 1}, \quad (115)$$

and find that

$$\lambda_1 = 1.2981, \lambda_2 = 0.1882, \lambda_3 = 0.9037 \quad (116)$$

To determine the total initial mass for each rocket, we use the optimal mass ratio:

$$\frac{m_{0(i)}}{m_L} = \prod_{i=1}^3 \left(\frac{\lambda_i + 1}{\lambda_i} \right) \quad (117)$$

We substitute Eq. 116 to find that

$$\frac{m_{0(i)}}{m_L} = 23.5462 \quad (118)$$

To calculate the total initial masses for each rocket, we plug in the respective values of m_L of 120,000 kg from the problem statement into Eq.118 and were able to find

$$m_{0(1)} = 2.873 * 10^6 \text{ kg} \quad (119)$$

These masses are recorded under “Total mass including payload” in Table 14.

The final step is to calculate the masses for each stage. To do this, we use another equation for payload ratio:

$$\lambda_i = \frac{m_{0(i+1)}}{m_{0(i)} - m_{0(i+1)}} \quad (120)$$

Knowing the payload ratio for each stage and having the mass of the payload for each architecture, we were then able to work our way down the rocket using Eq. 120 to find the overall rocket mass at the ignition of each stage

$$m_{0(3)} = m_L \left(\frac{1 + \lambda_3}{\lambda_3} \right) = 270,153.33 \text{ kg} \quad (121)$$

$$m_{0(2)} = m_{0(3)} \left(\frac{1+\lambda_2}{\lambda_2} \right) = 1,825,995.31 \text{ kg}$$

$$m_{0(1)} = m_{0(2)} \left(\frac{1+\lambda_1}{\lambda_1} \right) = 2,873,102.09 \text{ kg}$$

We can then use these values to find the mass of each stage

$$m_{3rd} = m_{0(3)} - m_L = 150,153.33 \text{ kg} \quad (122)$$

$$m_{2nd} = m_{0(2)} - m_{0(3)} = 1,555,837.98 \text{ kg}$$

$$m_{1st} = m_{0(1)} - m_{0(2)} = 1,047,106.78 \text{ kg}$$

Finally, we can calculate the dry mass of each stage ($m_{b(i)}$) using the following equation

$$R_i = \frac{M_{0(i)}}{M_{b(i)}} \quad (123)$$

Plugging in the appropriate values for R_i and $M_{0(i)}$ of each stage we find that the dry / burn out mass is

$$m_{b(1st)} = 1,657,498.66 \text{ kg} \quad (124)$$

$$m_{b(2nd)} = 346,815.27 \text{ kg}$$

$$m_{b(3rd)} = 145,229.89 \text{ kg}$$

Table 14: Mass of each stage for two architectures

Architecture	1st stage mass	2nd stage mass	3rd stage mass	Total mass including payload
1	2,418,772.43 kg	677,149.40 kg	N/A	3,215,921.92 kg
2	1,047,106.78 kg	1,555,837.98 kg	150,153.33 kg	2,873,102.09 kg

B. Rocket sizing

In this section, we estimate the required thrust levels for the launch vehicle, burn times, mass flow rates, and approximate number of engines to complete each stage. These estimates are informed by the thrust to weight requirements stipulated in Tables 11/12, the actual engine performance shown in Table 13, and the masses reported in Table 14.

For the thrust level at ignition (as reported in Table 3), we assume the first stage rockets will be at sea level and the upper stage rockets will both ignite in near vacuum like conditions. We also note that to calculate the actual total thrust of each stage, we first

determined the appropriate thrust value based on the required thrust to weight. We then determined the minimum number of engines necessary to provide this thrust (Table 4). We report the value of the thrust given by multiplying this number of engines by the reported thrust per engine. We estimate in this section as well as the diameter of the first stage for both architectures by determining the optimal circular diameter that would fit the indicated number of engines.

To estimate the thrust for each stage, we first calculate the initial mass of the overall rocket at the ignition of each stage. These can be inferred from Table 14. In particular, we find for the first architecture

$$m_{0(1)} = 2,418,772.43 \text{ kg}, m_{0(2)} = 677,149.40 \text{ kg} \quad (125)$$

For the second architecture

$$\begin{aligned} m_{0(3)} &= m_L \left(\frac{1+\lambda_3}{\lambda_3} \right) = 270,157.33 \text{ kg} \\ m_{0(2)} &= m_{0(3)} \left(\frac{1+\lambda_2}{\lambda_2} \right) = 1,825,995.31 \text{ kg} \\ m_{0(1)} &= m_{0(2)} \left(\frac{1+\lambda_1}{\lambda_1} \right) = 2,873,102.09 \text{ kg} \end{aligned} \quad (126)$$

The first step is to then determine the appropriate thrust value based on the required thrust to weight ratio (T/W)

$$\left(\frac{T}{W} \right)_i = \frac{T_i}{m_{0(i)} g_0} \quad (127)$$

We substitute Eq.125 and Eq.126 into Eq.127 using the $\left(\frac{T}{W} \right)_i$ values from Tables 11/12 to find values for the first architecture

$$T_1 = 35,592 \text{ kN}, T_2 = 5,314.3 \text{ kN} \quad (128)$$

and for the second architecture

$$T_1 = 33,822 \text{ kN}, T_2 = 16,122 \text{ kN}, T_3 = 1,590.1 \text{ kN} \quad (129)$$

To determine the minimum number of engines necessary for each stage, we divide the calculated T_i values by the reported thrust levels by the reported thrust levels from Table 13

$$(\# \text{ of engines})_i = \frac{T_i}{T_{JMT-1 \text{ or } JMT-2}} \quad (130)$$

where T_{JMT-1} denotes the thrust for the first stage JMT-1 engine at sea level and T_{JMT-2} is the thrust for the upper stage JMT-2 engines in near vacuum like conditions. We substitute Eq.128 and Eq.129 and use the reported thrust levels in Table 13 into Eq.130 to find the number of engines for each stage of the first architecture

$$(\# \text{ of engines})_1 \approx 6, (\# \text{ of engines})_2 \approx 6 \quad (131)$$

and of the second architecture

$$(\# \text{ of engines})_1 \approx 5, (\# \text{ of engines})_2 \approx 16, (\# \text{ of engines})_3 \approx 2 \quad (132)$$

To determine the thrust at ignition, we multiply the number of necessary engines by the reported thrust

$$T_{act(i)} = (\# \text{ of engines})_i * T_{(JMT-1 \text{ or } JMT-2)} \quad (133)$$

where $T_{act(i)}$ is the actual thrust at ignition at the “ith” stage.

We substitute Eq.131 and Eq.132 and use the reported thrust levels in Table 13 into Eq.133 to find actual thrust of each stage for the first architecture

$$T_{act(1)} = 41,478 \text{ kN}, T_{act(2)} = 6,067.8 \text{ kN} \quad (134)$$

and the second

$$T_{act(1)} = 34,565 \text{ kN}, T_{act(2)} = 18,203.4 \text{ kN}, T_{act(3)} = 2,022.6 \text{ kN} \quad (135)$$

To determine the flow rate of each stage, we use the equation for specific impulse:

$$I_{sp(JMT-1 \text{ or } JMT-2)} = \frac{T_{act(i)}}{\dot{m}_i g_0} \quad (136)$$

where \dot{m}_i is the flow rate of the “ith” stage, $I_{sp(JMT-1)}$ denotes the specific impulse for the first stage JMT-1 engines at sea level and $I_{sp(JMT-2)}$ is the specific impulse for the upper stage JMT-2 engines in near vacuum like conditions.

We then substitute Eq.134, Eq.135 and use the appropriate I_{sp} values from Table 13 into Eq.136 to find mass flow rate for the first architecture

$$\dot{m}_1 = 15,105.9 \text{ kg/s}, \dot{m}_2 = 2,093.9 \text{ kg/s} \quad (137)$$

and the second

$$\dot{m}_1 = 12,588.2 \text{ kg/s}, \dot{m}_2 = 3650.08 \text{ kg/s}, \dot{m}_3 = 521.44 \text{ kg/s} \quad (138)$$

To calculate the burn time of each stage, we first calculated the mass of propellant that would be in each stage using the following equation:

$$m_{p(i)} = m_{0(i)} - m_{b(i)} = m_{0(i)} \left(1 - \frac{1}{R_i}\right) \quad (139)$$

Aero 535

Jorns

Fall 2022

where $m_{p(i)}$ is the mass of propellant stored in each individual “ith” stage. Plugging in the values from Eq. (125) and Eq. (128) we find the propellant mass for the first architecture

$$m_{p(1)} = 1,500,700 \text{ kg}, m_{p(2)} = 496,210 \text{ kg} \quad (140)$$

and the second

$$m_{p(1)} = 995,710.8 \text{ kg}, m_{p(2)} = 1,447,256.72 \text{ kg}, m_{p(3)} = 121,651.95 \text{ kg} \quad (141)$$

We plugged in these values for propellant mass into the following equation to get burn time for each stage $t_{B(i)}$:

$$t_{B(i)} = \frac{m_{p(i)}}{\dot{m}_i} \quad (142)$$

By substituting Eq. (141) into Eq. (142) we found the following burn times for each stage of the first architecture

$$t_{B(1)} = 99.34 \text{ s}, t_{B(2)} = 236.98 \text{ s} \quad (143)$$

and the second

$$t_{B(1)} = 79.1 \text{ s}, t_{B(2)} = 396.5 \text{ s}, t_{B(3)} = 233.3 \text{ s} \quad (144)$$

Finally, to determine the stage diameters we use an online source that provides a table of encircling radius solutions for a two-dimensional circle packing problem [3]. We multiply the online solution for the designated number of circles by the diameter of each engine

$$d_{(i)} = S_{enc} * d_{engine(JMT-1 \text{ or } JMT-2)} \quad (145)$$

where $d_{(i)}$ denotes the diameter of each “ith” stage, S_{enc} is the encircling solution from the online source, $d_{engine(JMT-1)}$ is the diameter of the first stage JMT-1 engine, and $d_{engine(JMT-2)}$ is the diameter of the upper stage JMT-2 engines.

We then substitute the online solutions for the number of engines in a stage and the engine diameters in Table 13 for the first architecture

$$d_1 = 11.85 \text{ m}, d_2 = 11.65 \text{ m} \quad (146)$$

and the second

$$d_1 = 10.67 \text{ m}, d_2 = 10.67 \text{ m}, d_3 = 10.67 \text{ m} \quad (147)$$

Table 15: Engine characteristics for each architecture

Architecture	Stage	Thrust at ignition (kN)	Flow rate (kg/s)	Burn time (s)	Engine type	Number of engines	Stage diameter (m)
1 (2-Stage)	1 st	41,478	15,106	99.344	JMT-1	6	11.85
2 (3-Stage)	1 st	34,565	12,588	79.02	JMT-1	5	10.67
1	2 nd	6,067.8	2,093.9	236.98	JMT-2	6	11.65
2	2 nd	18,203.4	3650.08	396.5	JMT-2	16	10..67
2	3 rd	2,022.6	521.44	233.3	JMT-2	2	10.67

C. Staging Downselect

As seen from the analysis above, the second architecture with 3 stages provides a much more feasible alternative to the first architecture. It is clear from Table 16 that architecture 2 has a smaller overall mass by 342,819.83 kg and a smaller dry mass by 910,528.38 kg which are both significant margins. This indicates that there would be less engineering complexity in building the launch vehicle and would also help reduce cost. The second architecture does have more engines but this correlates to having a smaller engine diameter for the upper stages and is not expected to have a substantial impact on performance. Another advantage that was determined from our analysis is that the first stage is smaller in the second architecture as well in comparison to the first architecture.

Table 16: Spec Comparison Between Both Architectures

Specs	Architecture 1	Architecture 2
Overall mass	3,215,921.92 kg	2,873,102.09 kg
Total burn time	336.324 s	659.13 s
Propellant mass	1,996,910 kg	2,564,619.47 kg
Overall Dry Mass	1,219,011 kg	308,482.62 kg
# of Engines	12	23

VI. Launch trajectory

With the sizing of the vehicle complete, a 2D simulation of its launch can be performed. The simulation is done by numerically integrating a set of kinematic equations over a series of discrete time steps until it reaches the burnout time of $t = 670$ seconds. For our initial simulation, a series of simplifications are made

1. The initial part of the trajectory is the result of a gravity turn
2. After the gravity turn is complete, we gimbal the rocket thrust vector to provide a constant pitch angle, θ with respect to the local horizon
3. We neglect the rotation of the earth.
4. We neglect atmospheric drag and lift and solve for the trajectory in 2D
5. We assume the mass flow rate through the engines for each stage is constant during the burn (Table 7). However, the thrust level and specific impulse will change because the pressure is varying.
6. We target a destination altitude of 400 km by adjusting the gimbal angle of the thruster.

Table 17: Timing of Mission Events

Timing	Mission Event
$t + 0$ s	Liftoff
$t + 25$ s	Pitch maneuver 0.5° from vertical. Vehicle begins gravity turn
$t + 79.1$ s	Main engine cutoff (MECO) and first stage separation. 2nd stage ignition
$t + 151.58$ s	Vehicle begins first pitch maneuver at 34° from local horizon
$t + 475.60$ s	Second engine cutoff (SECO) and second stage separation. 3rd stage ignition
$t + 494.89$ s	Vehicle begins second pitch maneuver at -8° from local horizon
$t + 651.31$ s	Third engine cutoff. Orbit achieved

In order to simulate the launch up to orbit, a set of governing equations using kinematics and first principles are used. The equations are numerically integrated using a Δt value of 0.001 seconds for the simulation. As the vehicle ascends into orbit, its mass is reduced through the consumption of propellant in order to create thrust. Thus the first equation can simply be written as:

$$m_t = m_{t-\Delta t} - \Delta m_{stage} \quad (148)$$

where m_t is the mass at time t , $m_{t-\Delta t}$ is the mass at the timestep before t , and Δm_{stage} is the amount of mass being consumed for the current stage of the rocket over a given Δt . By substituting Eq.38, the acceleration of the vehicle can be determined:

$$a_t = \frac{T_{stage}}{m_t} * \cos(\theta - \psi_{t-\Delta t}) - g_0 \sin(\psi_{t-\Delta t}) \quad (149)$$

where the acceleration of the vehicle is a function of the thrust (T_{stage}) of the current stage, the mass (m_t) at the current timestep, the flight path angle ($\psi_{t-\Delta t}$) of the previous timestep, and gravity all help to determine the vehicle's acceleration. From the equation it is also evident that gravity has a large effect on the acceleration of the vehicle. This can be mitigated as much as possible though through the reduction of the flight path angle. Also note that the drag term has been taken out of the equation due to a simplification made.

From the Eq.35, the velocity of the vehicle can be determined.

$$v_t = v_{t-\Delta t} + a_t \cdot \Delta t \quad (150)$$

Here it can be seen in the equation that finding the velocity for the given time-step is as simple as adding the acceleration multiplied by the time-step size to the velocity of the previous time-step. This velocity is considered to be pointing in the direction of the flight path angle thus requiring no separation into its component v_x and v_y .

With the velocity and its direction now known, the change in altitude and downrange distance can be calculated using the following:

$$alt_t = alt_{t-\Delta t} + v_t \cdot \sin(\psi_{t-\Delta t}) \cdot \Delta t \quad (151)$$

$$xdis_t = xdis_{t-\Delta t} + v_t \cdot \cos(\psi_{t-\Delta t}) \cdot \Delta t \quad (152)$$

Similar to the velocity equations, the time-step values of altitude and downrange distance are calculated by adding the amount of the previous time-step to the velocity multiplied by sine or cosine for the change in altitude and downrange distance respectively.

With Eq.41 and Eq.42, the change in the flight path angle can then be calculated. For the first 20.49 seconds of the flight, the flight path angle remains constant in the vertical at 90° above the horizontal. At the timestep of $t + 25$ seconds after liftoff, the vehicle initiates a gravity turn by changing ψ to be 89.5° . With this change, the vehicle will begin a zero-lift trajectory until the pitch maneuver takes place. The equations used to calculate ψ is as follows:

$$\psi_t = \psi_{t-\Delta t} - \left[\left(\frac{g_0}{v_t} - \frac{v_t}{r_e + alt_t} \right) \cdot \cos(\psi_{t-\Delta t}) + \frac{a_t}{v_t} \cdot \sin(\theta - \psi_{t-\Delta t}) \right] \cdot \Delta t \quad (153)$$

where ψ_t is a function of gravity, velocity, and the distance from Earth's center and θ is the pitch angle. Up until the navigation program takes over, $\theta = \psi_{t-\Delta t}$ thus erasing that term from the equation. This is because until the navigation program takes over, θ is equal to $\psi_{t-\Delta t}$ because the flight path angle and the alignment of the vehicle are the same since it is a zero lift trajectory. With all of the main equations needed to simulate the launch, all that is left is refining the trajectory to ensure a proper orbital insertion.

Launch vehicles use a zero-lift trajectory in order to minimize any lateral forces on the vehicle. Being essentially a long Euler-column, they are not built to withstand excessive lateral loads induced by drag or lift on the structure thus necessitating a zero-lift trajectory. Once dynamic pressure begins to drop off significantly as the vehicle approaches the boundary of space, a zero-lift trajectory is no longer required. With this, a pitch angle can be induced on the vehicle to achieve the desired orbit. This

Aero 535

Jorns

Fall 2022

induced angle means that the vehicle's thrust is no longer in line with its velocity vector thus affecting the acceleration and flight path angles (Eq. 39 and Eq. 43 respectively) by giving them a pitch angle that is different from the flight path angle (whereas before they cancel). In order to achieve a proper orbit, two different pitch angles are used to refine the trajectory into the desired orbit. The first pitching maneuver is at 34° above the local horizon and takes place 151.58 seconds after liftoff. The second maneuver aims the vehicle 8° below the local horizon and occurs 494.89 seconds after liftoff.

Our final estimate for velocity and delivered mass is 178529.89 kg of which, 120,000 kg is payload. There is an estimated 30,000 kg of propellant (nearly a minute of burn time) leftover from the ascent which leaves around 700 m/s of delta-v for boosting the MAV further or for deorbiting the vehicle. It could also be used to further refine the vehicle and decrease the mass carried thus reducing the final cost and complexity. The final orbital velocity is 7.570 km/s at 400.44 km putting it within 0.4% of the target velocity and 0.11% of the target altitude.

The figures below illustrate the ascent of the vehicle (Figures 19 and 20) and the g-loading (Fig. 21). G-loading does begin to exceed allowable levels but can be mitigated via unmanned launches, throttling the vehicle, or potentially sacrificing an engine or two though this is likely the least optimal solution given the thrust requirements it has at ignition.

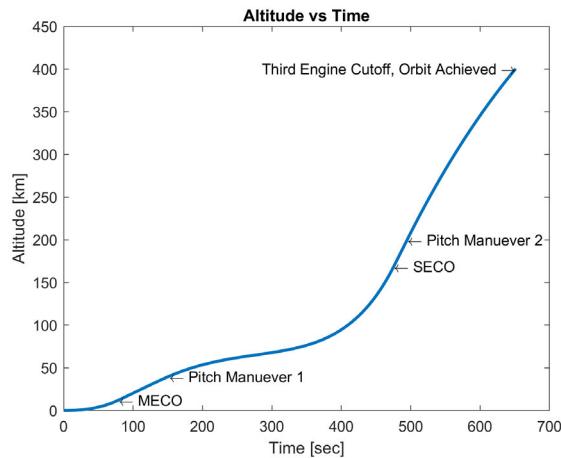


Figure 19: Altitude vs Time with Mission Events Noted

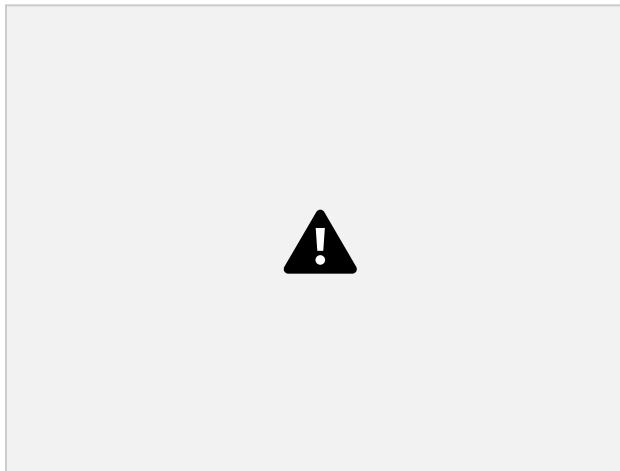


Figure 20: Altitude vs Downrange Distance with Mission Events Noted

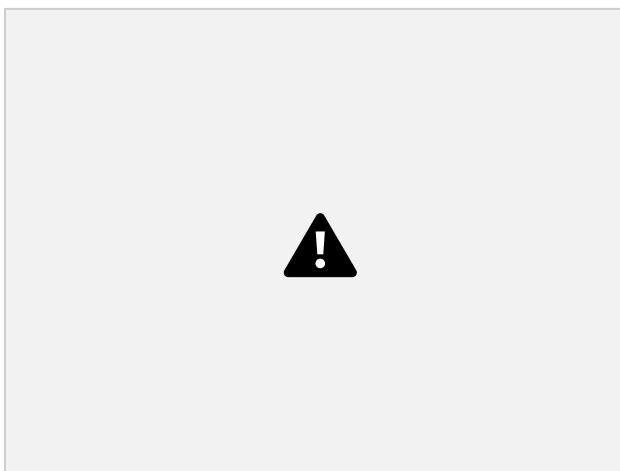


Figure 21: G-Loading vs Altitude with Mission Events Noted

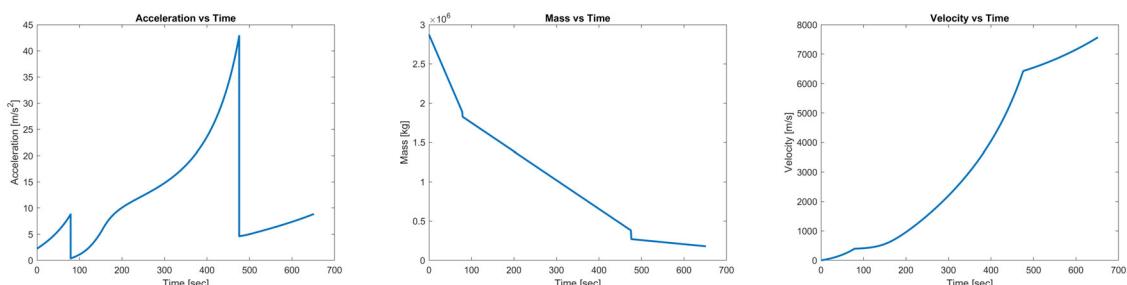


Figure 22: Additional Figures for Acceleration, Mass, and Velocity vs Time

VII. Conclusions and Discussion

For this proposal, we were given the task of creating a launch vehicle that would be able to launch a payload mass of 120,000 kg up to an altitude of 400km with a Δv of at least 9.3 km/s. To do this we decided to first design engines that could then be used in a rocket sizing study. We chose LOX/RP-1 to be our fuel mixture for our first stage engine (JMT-1), and

Aero 535

Jorns

Fall 2022

LOX/LH₂ to be the fuel mixture for the upper stage(s) engine (JMT-2). LOX/RP-1 was chosen as it is a widely available and high density fuel that has been widely used for rockets over the years, and it is liquid at room temperature which aids in reducing complexity of the first stage which will already have to have a considerable mass to support and lift the stages above it. LOX/LH₂ was chosen for its high performance compared to other fuels and its clean products.

With these two fuel mixtures being used for the upper and lower stages we decided to base initial assumptions of the engines off of Saturn V's F-1 and J-2 engines as they both were incredibly powerful engines that proved to have great success over the years. We used CEA to first refine our own O/F ratio and then our own expansion ratio to attempt to maximize performance as much as we could. Using both shifting equilibrium and frozen flow analysis in these steps, we were able to find the performance bounds of our engine, which we then carried over and refined in the nozzle design. We chose to design our nozzle using the rao method for designing bell nozzles, as bell nozzles are the most commonly used nozzles in rocket engines. We first carried out a study to define the normalized length of our shortened bell nozzle and found 80% to be the optimal percentage in gaining a maximum performance while not creating an overly big and heavy engine (which would drive up the total size and mass of our launch vehicle). Here, we also defined our performance to be 75% of the way up from the lower bound of frozen flow analysis to the upper bound of shifting equilibrium. This was done to help us finalize some initial performance parameters, and we knew from past experiences that the actual performance was closer to the shifting equilibrium estimate than the frozen flow estimate. From this estimate and our bell nozzle design, we were able to extract the following performance metrics:

Table 18: Final Engine Performance Metrics

Engine	Propellant	Sea level thrust (kN)	Vacuum thrust (kN)	SL Isp (s)	Vacuum Isp (s)	Engine diameter	Mass Flow Rate (kg/s)
JMT-1	LOX/RP-1	6,913 kN	7,341.7 kN	279.9 s	297.3 s	3.95 m	2,517.8
JMT-2	LOX/LH ₂	981.7 kN	1,011.3 kN	383.8 s	395.4s	1.70 m	260.7 kg/s

Taking these performance calculations, we can now compare how our JMT-1 and JMT-2 engines fare against the F-1 and J-2 engines that they were based off of.

Table 19: Baseline Engines Comparison

Engine	Propellant	Sea level thrust (kN)	Vacuum thrust (kN)	SL Isp (s)	Vacuum Isp (s)	Engine diameter	Mass Flow Rate (kg/s)
F-1	LOX/RP-1	6,770 kN	7,770 kN	263 s	304 s	3.7 m	2,577
J-2	LOX/LH ₂	486.2 kN	1,033.1 kN	200 s	421 s	2.1 m	240.7 kg/s

Aero 535

Jorns

Fall 2022

With this comparison, we can see that we succeeded in creating a first stage engine more powerful in thrust at sea-level than the F-1, while also having slightly less mass flow rate. However, our JMT-2 engine fell a bit short of beating out the J-2 engine in performance at vacuum conditions. We then moved on to designing the injectors and pumps that would make these engines function the way that they were designed to.

To design the injectors of each engine, we needed to consider which configurations worked well with our chosen propellants and were able to achieve high combustion efficiency. An orifice type and diameter were selected first to begin the general calculations of injector sizing. For JMT-1, we chose a pintle as this type is compatible with RP-1 and also only requires one main element that is located in the center of the injector plate to allow for simplicity. Our analysis indicated that roughly 6484 oxidizer orifices were needed for the necessary flow rate along a pintle diameter of 0.86 m. For JMT-1, our team chose a coaxial post injector that is specifically designed to work well with cryogenic propellants like LOX/LH₂ after being gasified by regenerative heating. Through an iterative process of changing the width of the fuel annulus in order to match the number of oxidizer injectors to the number of fuel injectors, it was found that a width of 1.1 mm was sufficient. We also were able to determine that there would need to be 1309 coaxial posts on the face of the injector.

Each engine was also designed with a pump-fed system that relied on a turbopump to generate flow and increase propellant pressure to the chamber values. This kind of machinery consists of a turbine that is powered by expanded hot gas and a pump side that typically consist of a centrifugal, rotating surface. The JMT-1 was chosen to have a gas generator, similar to the F-1 engine. This particular system is simple and can support high pressure increases across pumps and therefore also higher chamber pressure. Our analysis found that it takes 8.84 MW and 19.75 MW to power the fuel and oxidizer pumps, respectively. As for JMT-2, we chose an expander cycle as this system works with heated gas. It also has the advantage of being a “closed cycle” where all propellant is sent through the combustion chamber and yields a higher specific impulse, a relevant criteria for our upper stages. Our analysis found that it takes 4.63 MW and 1.15 MW to power the fuel and oxidizer pumps, respectively. We believe the fuel value is higher due to the vast density disparity between LOX and LH₂.

With our engines now fully designed, we moved onto the launch vehicle design. Here we started with a trade between two different architectures to get us to the target orbit, one being a two-stage launch vehicle and the other being a 3-stage launch vehicle. Working our way down from the payload mass, and making a few assumptions about the coefficients needed to size up the different stages, we were able to back out the masses of each stage of both architectures (Table 14). This helped us to find how much thrust would be needed per stage, and therefore how many of the JMT-1 and JMT-2 engines would be needed. In the end, the 3-stage launch vehicle was our pick as it had a much lower dry mass that led to the overall launch vehicle being significantly lighter than the 2-stage launch vehicle. This led us to conclude that the 3-stage would be a much more feasible option, and also matches the set-up of the Saturn V which used the engines we based our engines off of.

Given the amount of delta-V leftover in the 3rd stage, the design could benefit from optimization or could use said delta-V to boost the MAV further or deorbit the vehicle. That said, the vehicle was placed with high accuracy into its intended orbit. Further refinements to the pitch maneuvers or gravity turn initiation could see this accuracy increase though it is suitable for now. As seen in Figure 21, the G-loading does begin to exceed the levels that NASA is willing to allow its astronauts to tolerate however this lasts for roughly 40 seconds and could be allowable if there are no astronauts onboard or could be mitigated via throttling down the engines when the g-forces begins to exceed 3 g's.

VIII. Appendix

IX. References

- [1] Ivchenkov, G., "Evaluation of F-1 characteristics, based on the analysis of heat transfer and strength of the tubular cooling jacket," *Aulis* Available: https://www.aulis.com/PDF/F-1_Evaluation.pdf
- [2] Deeken, J., et al., "Multidisciplinary Design Optimization of Reusable Launch Vehicles for Different Propellants and Objectives," Sep. 2020. <https://arxiv.org/pdf/2009.01664.pdf>
- [3] "Circle packing in a circle," *Wikipedia* Available: https://en.wikipedia.org/wiki/Circle_packing_in_a_circle
- [4] Sutton, G. P., and Biblarz, O., *Rocket Propulsion Elements*, Hoboken, NJ: John Wiley & Sons Inc., 2017.

X. Matlab Code

CEA Analysis Code:

535 Final Project

```
% Jacob D., Maria M., Timothee G.

clc;
clear;

Fuel A (LOX/RP-1) Analysis 1 (O/F Ratio)

clc; clear;

% Set up and run CEA shifting equilibrium analysis with initial assumptions
% Output tabulated values into Excel
% Open Excel sheet here and extract values

CEA = readtable('FinalProjectCEAShifting1.xlsx');

CEA_A1 = table2array(CEA);

CEA = readtable('FinalProjectCEAFrozen1.xlsx');

CEA_A2 = table2array(CEA);

OF = CEA_A1(:,1); % O/F Ratio
```

Aero 535

Jorns

Fall 2022

```
gamma = CEA_A1(:,2); % Ratio of Specific Heats
M_bar = CEA_A1(:,3); % Molecular Weight [g/mol]
t = CEA_A1(:,4); % Combustion Temperature [K]
P = CEA_A1(:,5); % Combustion Pressure [...]
Isp = CEA_A1(:,6); % Specific Impulse [s]
Mach = CEA_A1(:,7); % Mach Number
Cf = CEA_A1(:,8); % Coefficient of Thrust
```

```
OF_fz = CEA_A2(:,1); % O/F Ratio
gamma_fz = CEA_A2(:,2); % Ratio of Specific Heats
M_bar_fz = CEA_A2(:,3); % Molecular Weight [g/mol]
t_fz = CEA_A2(:,4); % Combustion Temperature [K]
P_fz = CEA_A2(:,5); % Combustion Pressure [...]
Isp_fz = CEA_A2(:,6); % Specific Impulse [s]
Mach_fz = CEA_A2(:,7); % Mach Number
Cf_fz = CEA_A2(:,8); % Coefficient of Thrust
```

```
% Knowns and other parameters
P_a = 101325; % [Pa] Ambient Pressure at Sea Level
R_bar = 8.3145; % [J/mol*K] Universal Gas Constant
A_t = 0.4072; % [m^2] Throat Area
c_f = 0.95; % Thrust Thrust Correction Factor
c_d = 1.1; % Discharge Correction Factor
g0 = 9.81; % [m/s^2] Gravitational Constant
q=0;
w=0;
e=0;
```

```
for i =1:63
    if (i==1) || (i==4) || (i==7) || (i==10) || (i==13) || (i==16) || (i==19) || (i==22) ||
(i==25) || (i==28) || (i==31) || (i==34) || (i==37) || (i==40) || (i==43) %|| (i==46) ||
(i==49) || (i==52) || (i==55) || (i==58) || (i==61)
        q = q+1;
        OF_comb(1,q) = OF(i,1);
        gamma_comb(1,q) = gamma(i,1);
        M_bar_comb(1,q) = M_bar(i,1)/1000; %[kg/mol]
        t_comb(1,q) = t(i,1); % [K]
        P_comb(1,q) = P(i,1)*1E5; % [Pa]
        Isp_comb(1,q) = Isp(i,1)/g0;
```

```

Mach_comb(1,q) = Mach(i,1);
Cf_comb(1,q) = Cf(i,1);

OF_comb_fz(1,q) = OF_fz(i,1);
gamma_comb_fz(1,q) = gamma_fz(i,1);
M_bar_comb_fz(1,q) = M_bar_fz(i,1)/1000; %[kg/mol]
t_comb_fz(1,q) = t_fz(i,1); % [K]
P_comb_fz(1,q) = P_fz(i,1)*1E5; % [Pa]
Isp_comb_fz(1,q) = Isp_fz(i,1)/g0;
Mach_comb_fz(1,q) = Mach_fz(i,1);
Cf_comb_fz(1,q) = Cf_fz(i,1);

elseif (i==2) || (i==5) || (i==8) || (i==11) || (i==14) || (i==17) || (i==20) || (i==23)
|| (i==26) || (i==29) || (i==32) || (i==35) || (i==38) || (i==41) || (i==44) %|| (i==47) ||
(i==50) || (i==53) || (i==56) || (i==59) || (i==62)

w = w+1;

OF_t(1,w) = OF(i,1);
gamma_t(1,w) = gamma(i,1);
M_bar_t(1,w) = M_bar(i,1)/1000; %[kg/mol]
t_t(1,w) = t(i,1); % [K]
P_t(1,w) = P(i,1)*1E5; % [Pa]
Isp_t(1,w) = Isp(i,1)/g0;
Mach_t(1,w) = Mach(i,1);
Cf_t(1,w) = Cf(i,1);

OF_t_fz(1,w) = OF_fz(i,1);
gamma_t_fz(1,w) = gamma_fz(i,1);
M_bar_t_fz(1,w) = M_bar_fz(i,1)/1000; %[kg/mol]
t_t_fz(1,w) = t_fz(i,1); % [K]
P_t_fz(1,w) = P_fz(i,1)*1E5; % [Pa]
Isp_t_fz(1,w) = Isp_fz(i,1)/g0;
Mach_t_fz(1,w) = Mach_fz(i,1);
Cf_t_fz(1,w) = Cf_fz(i,1);

elseif (i==3) || (i==6) || (i==9) || (i==12) || (i==15) || (i==18) || (i==21) || (i==24)
|| (i==27) || (i==30) || (i==33) || (i==36) || (i==39) || (i==42) || (i==45) %|| (i==48) ||
(i==51) || (i==54) || (i==57) || (i==60) || (i==63)

e = e+1;

OF_e(1,e) = OF(i,1);
gamma_e(1,e) = gamma(i,1);
M_bar_e(1,e) = M_bar(i,1)/1000; %[kg/mol]
t_e(1,e) = t(i,1); % [K]

```

```

P_e(1,e) = P(i,1)*1E5; % [Pa]
Isp_e(1,e) = Isp(i,1)/g0;
Mach_e(1,e) = Mach(i,1);
Cf_e(1,e) = Cf(i,1);

OF_e_fz(1,e) = OF_fz(i,1);
gamma_e_fz(1,e) = gamma_fz(i,1);
M_bar_e_fz(1,e) = M_bar_fz(i,1)/1000; %[kg/mol]
t_e_fz(1,e) = t_fz(i,1); % [K]
P_e_fz(1,e) = P_fz(i,1)*1E5; % [Pa]
Isp_e_fz(1,e) = Isp_fz(i,1)/g0;
Mach_e_fz(1,e) = Mach_fz(i,1);
Cf_e_fz(1,e) = Cf_fz(i,1);

end
end

% Calculate Performance Metrics
T_se = (c_f/1000) .* (Cf_e .* P_comb .* A_t);
T_fz = (c_f/1000) .* (Cf_e_fz .* P_comb_fz .* A_t);
Isp_se = (c_f/c_d) .* (Isp_e);
Isp_fz = (c_f/c_d) .* (Isp_e_fz);

tiledlayout (1,3)
nexttile
a1 = plot(OF_e,T_se);
hold on
a2 = plot(OF_e,T_fz);
hold off
title('O/F Ratio vs Thrust')
xlabel('O/F Ratio')
ylabel('Thrust (kN)')
legend ([a1,a2],["Shifting Equilibrium","Frozen Flow"],'FontSize',6)
nexttile
a3 = plot(OF_e,Isp_se);
hold on
a4 = plot(OF_e,Isp_fz);
hold off
title('O/F Ratio vs Isp')

```

Aero 535

Jorns

Fall 2022

```
xlabel('O/F Ratio')
ylabel('Isp (s)')
legend ([a3,a4],["Shifting Equilibrium","Frozen Flow"],'FontSize',6)
nexttile
a5 = plot(OF_e,t_comb);
hold on
a6 = plot(OF_e,t_comb_fz);
hold off
title('O/F Ratio vs T0')
xlabel('O/F Ratio')
ylabel('T0 (K)')
legend ([a5,a6],["Shifting Equilibrium","Frozen Flow"],'FontSize',6)
```

Fuel A (LOX/RP-1) Analysis 2 (Expansion Ratio)

```
clc; clear;

% Now that the OF Ratio has been selected (3) we start varying the
% expansion ratio

% With the previous assumption being 16 we will now vary between 10 and 50

% Output tabulated values into Excel

% Open Excel sheet here and extract values

CEA = readtable('FinalProjectCEAShifting2.xlsx');
CEA_A3 = table2array(CEA);

CEA = readtable('FinalProjectCEAFrozen2.xlsx');
CEA_A4 = table2array(CEA);

Ae_At = CEA_A3(:,1); % Combustion Chamber Pressure (psi)
gamma = CEA_A3(:,2); % Ratio of Specific Heats
M_bar = CEA_A3(:,3); % Molecular Weight [g/mol]
t = CEA_A3(:,4); % Combustion Temperature [K]
P = CEA_A3(:,5); % Combustion Pressure [bar]
Isp = CEA_A3(:,6); % Specific Impulse [s]
Mach = CEA_A3(:,7); % Mach Number
Cf = CEA_A3(:,8); % Coefficient of Thrust
density = CEA_A3(:,9); % Density

Ae_At_fz = CEA_A4(:,1); % Combustion Chamber Pressure [psi]
gamma_fz = CEA_A4(:,2); % Ratio of Specific Heats
M_bar_fz = CEA_A4(:,3); % Molecular Weight [g/mol]
```

Aero 535

Jorns

Fall 2022

```
t_fz = CEA_A4(:,4); % Combustion Temperature [K]
P_fz = CEA_A4(:,5); % Combustion Pressure [bar]
Isp_fz = CEA_A4(:,6); % Specific Impulse [s]
Mach_fz = CEA_A4(:,7); % Mach Number
Cf_fz = CEA_A4(:,8); % Coefficient of Thrust
density_fz = CEA_A4(:,9); % Density
```

```
% Knowns and other parameters
P_a = 101325; % [Pa] Ambient Pressure at Sea Level
R_bar = 8.3145; % [J/mol*K] Universal Gas Constant
A_t = 0.4072; % [m^2] Throat Area
c_f = 0.95; % Thrust Thrust Correction Factor
c_d = 1.1; % Discharge Correction Factor
g0 = 9.81; % [m/s^2] Gravitational Constant
j=0;
```

```
for i =1:15
    if i==1
        Ae_At_comb = Ae_At(i,1);
        gamma_comb = gamma(i,1);
        M_bar_comb = M_bar(i,1); %[kg/kmol]
        t_comb = t(i,1); % [K]
        P_comb = P(i,1)*1E5; % [Pa]
        Isp_comb = Isp(i,1)/g0;
        Mach_comb = Mach(i,1);
        Cf_comb = Cf(i,1);
        density_comb = density(i,1);
```

```
Ae_At_comb_fz = Ae_At_fz(i,1);
gamma_comb_fz = gamma_fz(i,1);
M_bar_comb_fz = M_bar_fz(i,1); %[kg/kmol]
t_comb_fz = t_fz(i,1); % [K]
P_comb_fz = P_fz(i,1)*1E5; % [Pa]
Isp_comb_fz = Isp_fz(i,1)/g0;
Mach_comb_fz = Mach_fz(i,1);
Cf_comb_fz = Cf_fz(i,1);
density_comb_fz = density_fz(i,1);

elseif i==2
```

```

Ae_At_t = Ae_At(i,1);
gamma_t = gamma(i,1);
M_bar_t = M_bar(i,1); %[kg/kmol]
t_t = t(i,1); % [K]
P_t = P(i,1)*1E5; % [Pa]
Isp_t = Isp(i,1)/g0;
Mach_t = Mach(i,1);
Cf_t = Cf(i,1);
density_t = density(i,1);

```

```

Ae_At_t_fz = Ae_At_fz(i,1);
gamma_t_fz = gamma_fz(i,1);
M_bar_t_fz = M_bar_fz(i,1); %[kg/kmol]
t_t_fz = t_fz(i,1); % [K]
P_t_fz = P_fz(i,1)*1E5; % [Pa]
Isp_t_fz = Isp_fz(i,1)/g0;
Mach_t_fz = Mach_fz(i,1);
Cf_t_fz = Cf_fz(i,1);
density_t_fz = density_fz(i,1);

elseif i>2

```

```

j = j+1;
Ae_At_e(1,j) = Ae_At(i,1);
gamma_e(1,j) = gamma(i,1);
M_bar_e(1,j) = M_bar(i,1); %[kg/kmol]
t_e(1,j) = t(i,1); % [K]
P_e(1,j) = P(i,1)*1E5; % [Pa]
Isp_e(1,j) = Isp(i,1)/g0;
Mach_e(1,j) = Mach(i,1);
Cf_e(1,j) = Cf(i,1);
density_e(1,j) = density(i,1);

```

```

Ae_At_e_fz(1,j) = Ae_At_fz(i,1);
gamma_e_fz(1,j) = gamma_fz(i,1);
M_bar_e_fz(1,j) = M_bar_fz(i,1); %[kg/kmol]
t_e_fz(1,j) = t_fz(i,1); % [K]
P_e_fz(1,j) = P_fz(i,1)*1E5; % [Pa]
Isp_e_fz(1,j) = Isp_fz(i,1)/g0;
Mach_e_fz(1,j) = Mach_fz(i,1);

```

```

Cf_e_fz(1,j) = Cf_fz(i,1);
density_e_fz(1,j) = density_fz(i,1);

end
end

% Calculate Performance Metrics

T_se = (c_f/1000) .* (Cf_e .* P_comb .* A_t);
T_fz = (c_f/1000) .* (Cf_e_fz .* P_comb_fz .* A_t);
Isp_se = (c_f/c_d) .* (Isp_e);
Isp_fz = (c_f/c_d) .* (Isp_e_fz);
mw0_eq = M_bar_comb;
mw0_f = M_bar_comb_fz;
P0 = 10E6; % [Pa] Combustion Pressure
density0_eq = (mw0_eq.*P0)./(t_comb.*R_bar);
density0_f = (mw0_f.*P0)./(t_comb_fz.*R_bar);
density_ratio_eq = density_e./density0_eq;
density_ratio_f = density_e_fz./density0_f;
pressure_ratio_eq = P_e./P0;
pressure_ratio_f = P_e_fz./P0;
temp_ratio_eq = t_e./t_comb;
temp_ratio_f = t_e_fz./t_comb_fz;

tiledlayout (2,3)
nexttile
a1 = plot(Ae_At_e,T_se);
hold on
a2 = plot(Ae_At_e_fz,T_fz);
hold off
title('Expansion Ratio vs Thrust')
xlabel('Expansion Ratio')
ylabel('Thrust (kN)')
legend ([a1,a2],["Shifting Equilibrium","Frozen Flow"],'FontSize',6)

nexttile
a3 = plot(Ae_At_e,Isp_se);
hold on
a4 = plot(Ae_At_e_fz,Isp_fz);
hold off

```

Aero 535

Jorns

Fall 2022

```
title('Expansion Ratio vs Isp')
xlabel('Expansion Ratio')
ylabel('Isp (s)')
legend ([a3,a4],["Shifting Equilibrium","Frozen Flow"],'FontSize',6)
```

```
nexttile
a5 = plot(Ae_At_e,temp_ratio_eq,'LineWidth',2);
hold on
a6 = plot(Ae_At_e_fz,temp_ratio_f,'LineWidth',2);
hold off
title('Expansion Ratio vs. Temperature Ratio')
xlabel('Area Ratio')
ylabel('T/T0')
legend ([a5,a6],["Shifting Equilibrium","Frozen Flow"],'FontSize',6)
```

```
nexttile
a7 = plot(Ae_At_e,M_bar_e);
hold on
a8 = plot(Ae_At_e_fz,M_bar_e_fz);
hold off
title('Expansion Ratio vs Molecular Weight')
xlabel('Expansion Ratio')
ylabel('Molecular Weight (kg/mol)')
legend ([a7,a8],["Shifting Equilibrium","Frozen Flow"],'FontSize',6)
```

```
nexttile
a7 = plot(Ae_At_e,pressure_ratio_eq);
hold on
a8 = plot(Ae_At_e_fz,pressure_ratio_f);
hold off
title('Expansion Ratio vs. Pressure Ratio')
xlabel('Expansion Ratio')
ylabel('P/P0')
legend ([a7,a8],["Shifting Equilibrium","Frozen Flow"],'FontSize',6)
```

```
nexttile
a7 = plot(Ae_At_e,gamma_e);
hold on
```

Aero 535

Jorns

Fall 2022

```
a8 = plot(Ae_At_e_fz,gamma_e_fz);

hold off

title('Expansion Ratio vs. Specific Heat Ratio')

xlabel('Expansion Ratio')

ylabel('Specific Heat Ratio')

legend ([a7,a8],["Shifting Equilibrium","Frozen Flow"],'FontSize',6)
```

Fuel B Analysis 1 (O/F Ratio)

```
clc; clear;

% Set up and run CEA shifting equilibrium analysis with initial assumptions

% Output tabulated values into Excel

% Open Excel sheet here and extract values

CEA = readtable('FinalProjectCEAShifting3.xlsx');

CEA_A1 = table2array(CEA);

CEA = readtable('FinalProjectCEAFrozen3.xlsx');

CEA_A2 = table2array(CEA);
```

```
OF = CEA_A1(:,1); % O/F Ratio

gamma = CEA_A1(:,2); % Ratio of Specific Heats

M_bar = CEA_A1(:,3); % Molecular Weight [g/mol]

t = CEA_A1(:,4); % Combustion Temperature [K]

P = CEA_A1(:,5); % Combustion Pressure [...]

Isp = CEA_A1(:,6); % Specific Impulse [s]

Mach = CEA_A1(:,7); % Mach Number

Cf = CEA_A1(:,8); % Coefficient of Thrust
```

```
OF_fz = CEA_A2(:,1); % O/F Ratio

gamma_fz = CEA_A2(:,2); % Ratio of Specific Heats

M_bar_fz = CEA_A2(:,3); % Molecular Weight [g/mol]

t_fz = CEA_A2(:,4); % Combustion Temperature [K]

P_fz = CEA_A2(:,5); % Combustion Pressure [...]

Isp_fz = CEA_A2(:,6); % Specific Impulse [s]

Mach_fz = CEA_A2(:,7); % Mach Number

Cf_fz = CEA_A2(:,8); % Coefficient of Thrust
```

```
% Knowns and other parameters

P_a = 101325; % [Pa] Ambient Pressure at Sea Level

R_bar = 8.3145; % [J/mol*K] Universal Gas Constant
```

```

A_t = 0.1092; % [m^2] Throat Area
c_f = 0.95; % Thrust Thrust Correction Factor
c_d = 1.1; % Discharge Correction Factor
g0 = 9.81; % [m/s^2] Gravitational Constant
q=0;
w=0;
e=0;

for i =1:63

    if (i==1) || (i==4) || (i==7) || (i==10) || (i==13) || (i==16) || (i==19) || (i==22) ||
(i==25) || (i==28) || (i==31) || (i==34) || (i==37) || (i==40) || (i==43) %|| (i==46) ||
(i==49) || (i==52) || (i==55) || (i==58) || (i==61)

        q = q+1;

        OF_comb(1,q) = OF(i,1);

        gamma_comb(1,q) = gamma(i,1);

        M_bar_comb(1,q) = M_bar(i,1)/1000; %[kg/mol]

        t_comb(1,q) = t(i,1); % [K]

        P_comb(1,q) = P(i,1)*1E5; % [Pa]

        Isp_comb(1,q) = Isp(i,1)/g0;

        Mach_comb(1,q) = Mach(i,1);

        Cf_comb(1,q) = Cf(i,1);

    else

        OF_comb_fz(1,q) = OF_fz(i,1);

        gamma_comb_fz(1,q) = gamma_fz(i,1);

        M_bar_comb_fz(1,q) = M_bar_fz(i,1)/1000; %[kg/mol]

        t_comb_fz(1,q) = t_fz(i,1); % [K]

        P_comb_fz(1,q) = P_fz(i,1)*1E5; % [Pa]

        Isp_comb_fz(1,q) = Isp_fz(i,1)/g0;

        Mach_comb_fz(1,q) = Mach_fz(i,1);

        Cf_comb_fz(1,q) = Cf_fz(i,1);

    elseif (i==2) || (i==5) || (i==8) || (i==11) || (i==14) || (i==17) || (i==20) || (i==23) ||
(i==26) || (i==29) || (i==32) || (i==35) || (i==38) || (i==41) || (i==44) %|| (i==47) ||
(i==50) || (i==53) || (i==56) || (i==59) || (i==62)

        w = w+1;

        OF_t(1,w) = OF(i,1);

        gamma_t(1,w) = gamma(i,1);

        M_bar_t(1,w) = M_bar(i,1)/1000; %[kg/mol]

        t_t(1,w) = t(i,1); % [K]

        P_t(1,w) = P(i,1)*1E5; % [Pa]

        Isp_t(1,w) = Isp(i,1)/g0;

    end
end

```

```

Mach_t(1,w) = Mach(i,1);
Cf_t(1,w) = Cf(i,1);

OF_t_fz(1,w) = OF_fz(i,1);
gamma_t_fz(1,w) = gamma_fz(i,1);
M_bar_t_fz(1,w) = M_bar_fz(i,1)/1000; %[kg/mol]
t_t_fz(1,w) = t_fz(i,1); % [K]
P_t_fz(1,w) = P_fz(i,1)*1E5; % [Pa]
Isp_t_fz(1,w) = Isp_fz(i,1)/g0;
Mach_t_fz(1,w) = Mach_fz(i,1);
Cf_t_fz(1,w) = Cf_fz(i,1);

elseif (i==3) || (i==6) || (i==9) || (i==12) || (i==15) || (i==18) || (i==21) || (i==24)
|| (i==27) || (i==30) || (i==33) || (i==36) || (i==39) || (i==42) || (i==45) %|| (i==48) ||
(i==51) || (i==54) || (i==57) || (i==60) || (i==63)

e = e+1;

OF_e(1,e) = OF(i,1);
gamma_e(1,e) = gamma(i,1);
M_bar_e(1,e) = M_bar(i,1)/1000; %[kg/mol]
t_e(1,e) = t(i,1); % [K]
P_e(1,e) = P(i,1)*1E5; % [Pa]
Isp_e(1,e) = Isp(i,1)/g0;
Mach_e(1,e) = Mach(i,1);
Cf_e(1,e) = Cf(i,1);

OF_e_fz(1,e) = OF_fz(i,1);
gamma_e_fz(1,e) = gamma_fz(i,1);
M_bar_e_fz(1,e) = M_bar_fz(i,1)/1000; %[kg/mol]
t_e_fz(1,e) = t_fz(i,1); % [K]
P_e_fz(1,e) = P_fz(i,1)*1E5; % [Pa]
Isp_e_fz(1,e) = Isp_fz(i,1)/g0;
Mach_e_fz(1,e) = Mach_fz(i,1);
Cf_e_fz(1,e) = Cf_fz(i,1);

end
end

% Calculate Performance Metrics
T_se = (c_f/1000) .* (Cf_e .* P_comb .* A_t);
T_fz = (c_f/1000) .* (Cf_e_fz .* P_comb_fz .* A_t);
Isp_se = (c_f/c_d) .* (Isp_e);

```

Aero 535

Jorns

Fall 2022

```
Isp_fz = (c_f/c_d) .* (Isp_e_fz);

tiledlayout (1,3)

nexttile
a1 = plot(OF_e,T_se);
hold on
a2 = plot(OF_e,T_fz);
hold off
title('O/F Ratio vs Thrust')
xlabel('O/F Ratio')
ylabel('Thrust (kN)')
legend ([a1,a2],["Shifting Equilibrium","Frozen Flow"],'FontSize',6)
nexttile
a3 = plot(OF_e,Isp_se);
hold on
a4 = plot(OF_e,Isp_fz);
hold off
title('O/F Ratio vs Isp')
xlabel('O/F Ratio')
ylabel('Isp (s)')
legend ([a3,a4],["Shifting Equilibrium","Frozen Flow"],'FontSize',6)
nexttile
a5 = plot(OF_e,t_comb);
hold on
a6 = plot(OF_e,t_comb_fz);
hold off
title('O/F Ratio vs T0')
xlabel('O/F Ratio')
ylabel('T0 (K)')
legend ([a5,a6],["Shifting Equilibrium","Frozen Flow"],'FontSize',6)
```

Fuel B (LOX/LH₂) Analysis 2 (Expansion Ratio)

```
clc; clear;
clear;

% Now that the OF Ratio has been selected (3) we start varying the
% expansion ratio

% With the previous assumption being 16 we will now vary between 10 and 50

% Output tabulated values into Excel
```

Aero 535

Jorns

Fall 2022

```
% Open Excel sheet here and extract values
CEA = readtable('FinalProjectCEAShifting4.xlsx');
CEA_A3 = table2array(CEA);
CEA = readtable('FinalProjectCEAFrozen4.xlsx');
CEA_A4 = table2array(CEA);

Ae_At = CEA_A3(:,1); % Combustion Chamber Pressure (psi)
gamma = CEA_A3(:,2); % Ratio of Specific Heats
M_bar = CEA_A3(:,3); % Molecular Weight [g/mol]
t = CEA_A3(:,4); % Combustion Temperature [K]
P = CEA_A3(:,5); % Combustion Pressure [bar]
Isp = CEA_A3(:,6); % Specific Impulse [s]
Mach = CEA_A3(:,7); % Mach Number
Cf = CEA_A3(:,8); % Coefficient of Thrust
density = CEA_A3(:,9); % Density

Ae_At_fz = CEA_A4(:,1); % Combustion Chamber Pressure [psi]
gamma_fz = CEA_A4(:,2); % Ratio of Specific Heats
M_bar_fz = CEA_A4(:,3); % Molecular Weight [g/mol]
t_fz = CEA_A4(:,4); % Combustion Temperature [K]
P_fz = CEA_A4(:,5); % Combustion Pressure [bar]
Isp_fz = CEA_A4(:,6); % Specific Impulse [s]
Mach_fz = CEA_A4(:,7); % Mach Number
Cf_fz = CEA_A4(:,8); % Coefficient of Thrust
density_fz = CEA_A4(:,9); % Density

% Knowns and other parameters
P_a = 101325; % [Pa] Ambient Pressure at Sea Level
R_bar = 8.3145; % [J/mol*K] Universal Gas Constant
A_t = 0.1092; % [m^2] Throat Area
c_f = 0.95; % Thrust Thrust Correction Factor
c_d = 1.1; % Discharge Correction Factor
g0 = 9.81; % [m/s^2] Gravitational Constant
j=0;

for i =1:28
    if i==1
        Ae_At_comb = Ae_At(i,1);
```

```

gamma_comb = gamma(i,1);

M_bar_comb = M_bar(i,1); %[kg/kmol]

t_comb = t(i,1); % [K]

P_comb = P(i,1)*1E5; % [Pa]

Isp_comb = Isp(i,1)/g0;

Mach_comb = Mach(i,1);

Cf_comb = Cf(i,1);

density_comb = density(i,1);



---


Ae_At_comb_fz = Ae_At_fz(i,1);

gamma_comb_fz = gamma_fz(i,1);

M_bar_comb_fz = M_bar_fz(i,1); %[kg/kmol]

t_comb_fz = t_fz(i,1); % [K]

P_comb_fz = P_fz(i,1)*1E5; % [Pa]

Isp_comb_fz = Isp_fz(i,1)/g0;

Mach_comb_fz = Mach_fz(i,1);

Cf_comb_fz = Cf_fz(i,1);

density_comb_fz = density_fz(i,1);

elseif i==2

Ae_At_t = Ae_At(i,1);

gamma_t = gamma(i,1);

M_bar_t = M_bar(i,1); %[kg/kmol]

t_t = t(i,1); % [K]

P_t = P(i,1)*1E5; % [Pa]

Isp_t = Isp(i,1)/g0;

Mach_t = Mach(i,1);

Cf_t = Cf(i,1);

density_t = density(i,1);



---


Ae_At_t_fz = Ae_At_fz(i,1);

gamma_t_fz = gamma_fz(i,1);

M_bar_t_fz = M_bar_fz(i,1); %[kg/kmol]

t_t_fz = t_fz(i,1); % [K]

P_t_fz = P_fz(i,1)*1E5; % [Pa]

Isp_t_fz = Isp_fz(i,1)/g0;

Mach_t_fz = Mach_fz(i,1);

Cf_t_fz = Cf_fz(i,1);

density_t_fz = density_fz(i,1);

```

```

elseif i>2
    j = j+1;
    Ae_At_e(1,j) = Ae_At(i,1);
    gamma_e(1,j) = gamma(i,1);
    M_bar_e(1,j) = M_bar(i,1); %[kg/kmol]
    t_e(1,j) = t(i,1); % [K]
    P_e(1,j) = P(i,1)*1E5; % [Pa]
    Isp_e(1,j) = Isp(i,1)/g0;
    Mach_e(1,j) = Mach(i,1);
    Cf_e(1,j) = Cf(i,1);
    density_e(1,j) = density(i,1);

    Ae_At_e_fz(1,j) = Ae_At_fz(i,1);
    gamma_e_fz(1,j) = gamma_fz(i,1);
    M_bar_e_fz(1,j) = M_bar_fz(i,1); %[kg/kmol]
    t_e_fz(1,j) = t_fz(i,1); % [K]
    P_e_fz(1,j) = P_fz(i,1)*1E5; % [Pa]
    Isp_e_fz(1,j) = Isp_fz(i,1)/g0;
    Mach_e_fz(1,j) = Mach_fz(i,1);
    Cf_e_fz(1,j) = Cf_fz(i,1);
    density_e_fz(1,j) = density_fz(i,1);

end
end

% Calculate Performance Metrics
T_se = (c_f/1000) .* (Cf_e .* P_comb .* A_t);
T_fz = (c_f/1000) .* (Cf_e_fz .* P_comb_fz .* A_t);
Isp_se = (c_f/c_d) .* (Isp_e);
Isp_fz = (c_f/c_d) .* (Isp_e_fz);
mw0_eq = M_bar_comb;
mw0_f = M_bar_comb_fz;
P0 = 5.3E6; % [Pa] Combustion Pressure
density0_eq = (mw0_eq.*P0)./(t_comb.*R_bar);
density0_f = (mw0_f.*P0)./(t_comb_fz.*R_bar);
density_ratio_eq = density_e./density0_eq;
density_ratio_f = density_e_fz./density0_f;
pressure_ratio_eq = P_e./P0;
pressure_ratio_f = P_e_fz./P0;

```

Aero 535

Jorns

Fall 2022

```
temp_ratio_eq = t_e./t_comb;
temp_ratio_f = t_e_fz./t_comb_fz;

tiledlayout (2,3)
nexttile
a1 = plot(Ae_At_e,T_se);
hold on
a2 = plot(Ae_At_e_fz,T_fz);
hold off
title('Expansion Ratio vs Thrust')
xlabel('Expansion Ratio')
ylabel('Thrust (kN)')
legend ([a1,a2],["Shifting Equilibrium","Frozen Flow"],'FontSize',6)
```

```
nexttile
a3 = plot(Ae_At_e,Isp_se);
hold on
a4 = plot(Ae_At_e_fz,Isp_fz);
hold off
title('Expansion Ratio vs Isp')
xlabel('Expansion Ratio')
ylabel('Isp (s)')
legend ([a3,a4],["Shifting Equilibrium","Frozen Flow"],'FontSize',6)
```

```
nexttile
a5 = plot(Ae_At_e,temp_ratio_eq,'LineWidth',2);
hold on
a6 = plot(Ae_At_e_fz,temp_ratio_f,'LineWidth',2);
hold off
title('Expansion Ratio vs. Temperature Ratio')
xlabel('Area Ratio')
ylabel('T/T0')
legend ([a5,a6],["Shifting Equilibrium","Frozen Flow"],'FontSize',6)
```

```
nexttile
a7 = plot(Ae_At_e,M_bar_e);
hold on
a8 = plot(Ae_At_e_fz,M_bar_e_fz);
```

Aero 535

Jorns

Fall 2022

```
hold off

title('Expansion Ratio vs Molecular Weight')

xlabel('Expansion Ratio')

ylabel('Molecular Weight (kg/mol)')

legend ([a7,a8],["Shifting Equilibrium","Frozen Flow"],'FontSize',6)
```

```
nexttile

a7 = plot(Ae_At_e,pressure_ratio_eq);

hold on

a8 = plot(Ae_At_e_fz,pressure_ratio_f);

hold off

title('Expansion Ratio vs. Pressure Ratio')

xlabel('Expansion Ratio')

ylabel('P/P0')

legend ([a7,a8],["Shifting Equilibrium","Frozen Flow"],'FontSize',6)
```

```
nexttile

a7 = plot(Ae_At_e,gamma_e);

hold on

a8 = plot(Ae_At_e_fz,gamma_e_fz);

hold off

title('Expansion Ratio vs. Specific Heat Ratio')

xlabel('Expansion Ratio')

ylabel('Specific Heat Ratio')

legend ([a7,a8],["Shifting Equilibrium","Frozen Flow"],'FontSize',6)
```

Final Performance Metrics of the two Engines

```
clc; clear;

clear;

% Now that all the design metrics have been established extract the
% performance metrics
```

```
% Knowns and other parameters

R_bar = 8.3145; % [J/mol*K] Universal Gas Constant

c_f = 0.95; % Thrust Thrust Correction Factor

c_d = 1.1; % Discharge Correction Factor

g0 = 9.81; % [m/s^2] Gravitational Constant
```

Aero 535

Jorns

Fall 2022

```
% Output tabulated values into Excel

% Open Excel sheet here and extract values

CEA = readtable('JMT-1Shifting.xlsx');

JMT_1S = table2array(CEA);

CEA = readtable('JMT-1Frozen.xlsx');

JMT_1F = table2array(CEA);

CEA = readtable('JMT-2Shifting.xlsx');

JMT_2S = table2array(CEA);

CEA = readtable('JMT-2Frozen.xlsx');

JMT_2F = table2array(CEA);

gamma_1S = JMT_1S(:,1); % Ratio of Specific Heats
M_bar_1S = JMT_1S(:,2)/1000; % Molecular Weight [g/mol]
t_1S = JMT_1S(:,3); % Combustion Temperature [K]
P_1S = JMT_1S(:,4)*1E5; % Combustion Pressure [bar]
Isp_1S = JMT_1S(:,5)/g0; % Specific Impulse [s]
Mach_1S = JMT_1S(:,6); % Mach Number
Cf_1S = JMT_1S(:,7); % Coefficient of Thrust
Ivac_1S = JMT_1S(:,8)/g0; % Vacuum Specific Impulse [s]

gamma_1F = JMT_1F(:,1); % Ratio of Specific Heats
M_bar_1F = JMT_1F(:,2)/1000; % Molecular Weight [g/mol]
t_1F = JMT_1F(:,3); % Combustion Temperature [K]
P_1F = JMT_1F(:,4)*1E5; % Combustion Pressure [bar]
Isp_1F = JMT_1F(:,5)/g0; % Specific Impulse [s]
Mach_1F = JMT_1F(:,6); % Mach Number
Cf_1F = JMT_1F(:,7); % Coefficient of Thrust
Ivac_1F = JMT_1F(:,8)/g0; % Vacuum Specific Impulse [s]

gamma_2S = JMT_2S(:,1); % Ratio of Specific Heats
M_bar_2S = JMT_2S(:,2)/1000; % Molecular Weight [g/mol]
t_2S = JMT_2S(:,3); % Combustion Temperature [K]
P_2S = JMT_2S(:,4)*1E5; % Combustion Pressure [bar]
Isp_2S = JMT_2S(:,5)/g0; % Specific Impulse [s]
Mach_2S = JMT_2S(:,6); % Mach Number
Cf_2S = JMT_2S(:,7); % Coefficient of Thrust
Ivac_2S = JMT_2S(:,8)/g0; % Vacuum Specific Impulse [s]
```

Aero 535

Jorns

Fall 2022

```
gamma_2F = JMT_2F(:,1); % Ratio of Specific Heats
M_bar_2F = JMT_2F(:,2)/1000; % Molecular Weight [g/mol]
t_2F = JMT_2F(:,3); % Combustion Temperature [K]
P_2F = JMT_2F(:,4)*1E5; % Combustion Pressure [bar]
Isp_2F = JMT_2F(:,5)/g0; % Specific Impulse [s]
Mach_2F = JMT_2F(:,6); % Mach Number
Cf_2F = JMT_2F(:,7); % Coefficient of Thrust
Ivac_2F = JMT_2F(:,8)/g0; % Vacuum Specific Impulse [s]
```

```
% Calculate Performance Metrics for JMT-1
A_t = 0.4072; % [m^2] Throat Area
A_e = A_t * 30; % [m^2] Throat Area
```

```
T1se_s1 = (c_f/1000) * (Cf_1S(3) * P_1S(1) * A_t)
Isp1se_s1 = (c_f/c_d) * (Isp_1S(3))
m_dot_1S = (T1se_s1)/(Isp1se_s1 * g0) * 1000
Isp1se_vac = (c_f/c_d) * (Ivac_1S(3))
T1se_vac = (Isp1se_vac * m_dot_1S * g0)/1000
```

```
T1fz_s1 = (c_f/1000) * (Cf_1F(3) * P_1F(1) * A_t)
Isp1fz_s1 = (c_f/c_d) * (Isp_1F(3))
m_dot_1F = (T1fz_s1)/(Isp1fz_s1 * g0) * 1000
Isp1fz_vac = (c_f/c_d) * (Ivac_1F(3))
T1fz_vac = (Isp1fz_vac * m_dot_1F * g0)/1000
```

```
% Calculate Performance Metrics for JMT-2
A_t = 0.1092; % [m^2] Throat Area
ER = 60; % Expansion Ratio
A_e = A_t * ER; % [m^2] Exit Plane Area
d_e = 2 * sqrt(A_e/pi); % [m] Exit Plane Diameter
```

```
T2se_s1 = (c_f/1000) * (Cf_2S(3) * P_2S(1) * A_t)
Isp2se_s1 = (c_f/c_d) * (Isp_2S(3))
m_dot_2S = (T2se_s1)/(Isp2se_s1 * g0) * 1000
Isp2se_vac = (c_f/c_d) * (Ivac_2S(3))
T2se_vac = (Isp2se_vac * m_dot_2S * g0)/1000
```

```
T2fz_s1 = (c_f/1000) * (Cf_2F(3) * P_2F(1) * A_t)
```

Aero 535

Jorns

Fall 2022

```
Isp2fz_s1 = (c_f/c_d) * (Isp_2F(3))

m_dot_2F = (T2fz_s1)/(Isp2fz_s1 * g0) * 1000

Isp2fz_vac = (c_f/c_d) * (Ivac_2F(3))

T2fz_vac = (Isp2fz_vac * m_dot_2F * g0)/1000
```

Nozzle Design Code:

Bell Nozzle Design (JMT-1)

```
% We can now start the design of our nozzle

% Rao's Method for Bell Nozzle Design

% Common Values

r_t = 0.72/2; % [m] Throat Radius

r_c = 0.4 * r_t % [m] Radius of Curvature

Ae_At = 30; % Expansion Ratio

r_e = r_t * sqrt(Ae_At); % Exit Plane Radius

P_amb = 101325; % [Pa] Ambient pressure at sea-level

P_vac = 0; % [Pa] Ambient pressure at vacuum

A_t = pi * (r_t^2);

A_e = A_t * 30;

j=0;
```

```
for i = 60:10:100

    if i == 60

        j=j+1;

        theta_i(1,j) = 46.5; % Initial parabola angle

        theta_f(1,j) = 13.5; % Final parabola angle

        lambda(1,j) = 0.972; % Nozzle correction factor

    elseif i == 70

        j=j+1;

        theta_i(1,j) = 44; % Initial parabola angle

        theta_f(1,j) = 10; % Final parabola angle

        lambda(1,j) = 0.98; % Nozzle correction factor

    elseif i == 80

        j=j+1;

        theta_i(1,j) = 40.5; % Initial parabola angle

        theta_f(1,j) = 8; % Final parabola angle

        lambda(1,j) = 0.9875; % Nozzle correction factor

    elseif i == 90

        j=j+1;

        theta_i(1,j) = 38.5; % Initial parabola angle
```

Aero 535

Jorns

Fall 2022

```
theta_f(1,j) = 6; % Final parabola angle
lambda(1,j) = 0.99125; % Nozzle correction factor

else
    j=j+1;
    theta_i(1,j) = 36.5; % Initial parabola angle
    theta_f(1,j) = 4.5; % Final parabola angle
    lambda(1,j) = 0.9925;% Nozzle correction factor
end

% JMT-1 Shifting Equilibrium Values
T1_slse = T1se_s1 .* lambda;
T1_vacse = T1se_vac .* lambda;

% JMT-1 Frozen Flow Values
T1_slfz = T1fz_s1 .* lambda;
T1_vacfz = T1fz_vac .* lambda;

% Assume actual values lie 75% of way up from frozen flow boundary to
% shifting equilibrium boundary and get final values for thrust
delta_T1sl = T1_slse - T1_slfz;
delta_Isp1sl = Isp1se_s1 - Isp1fz_s1;
T1_sl = T1_slfz + (0.75 .* delta_T1sl);
Isp1_s1 = Isp1fz_s1 + (0.75 * delta_Isp1sl);
delta_T1vac = T1_vacse - T1_vacfz;
delta_Isp1vac = Isp1se_vac - Isp1fz_vac;
T1_vac = T1_vacfz + (0.75 .* delta_T1vac);
Isp1_vac = Isp1fz_vac + (0.75 * delta_Isp1vac);

i = 60:10:100
figure
a20 = plot(i,T1_sl);
hold on
a21 = plot(i,T1_vac);
hold off
title('Thrust vs. Normalized Length Percentage')
ylabel('Thrust (kN)')
xlabel('Normalized Length (%)')
legend ([a20,a21],["Sea-Level Thrust","Vacuum Thrust"],'FontSize',6)
```

Aero 535

Jorns

Fall 2022

```
% Picking a normalized length of 80% re-calculate final performance metrics
% of the JMT-1 Engine

T1sl = T1_s1(3)
T1vac = T1_vac(3)
Isp1sl = Isp1_s1
Isp1vac = Isp1_vac
m_dot_1sl = T1sl / (Isp1sl * g0) * 1000
m_dot_1vac = T1vac / (Isp1vac * g0) * 1000
r_e = sqrt(A_e/pi)
d_e = r_e * 2

% Plot the expansion region of the nozzle
x1a = r_c * sind(theta_i(3)); % [m] Initial x-coordinate of expansion region of nozzle
y1a = r_t + (r_c * (1-cosd(theta_i(3)))); % [m] Initial y-coordinate of expansion region of nozzle
r_e = r_t * sqrt(Ae_At); % [m] Radius of the exit plane
L_15 = ((r_t*(sqrt(Ae_At)-1)) + (r_c*((1/cosd(15))-1))) / (tand(15)) % [m] Length of nozzle before shortening
L = 0.8 * L_15 % [m] Length of normalized length nozzle
T_L = T1sl/L % [N/m] Thrust to Length Ratio
% Plot the initial radius of curvature part
x_c = 0:0.0001:x1a;
for i = 1:length(x_c)
    y_c(i) = (r_t + r_c) - sqrt(((r_c)^2) - ((x_c(i))^2));
end
plot ((x_c/r_t),(y_c/r_t))
hold on
rt = 0.36;
r_curve = 1.25*rt;
l_c = 1.02/2;
theta = 45;
Ac = 2*0.407;
r_comb = sqrt(Ac/pi);
x_start = (l_c - r_curve*(sind(theta)));
x_c3 = x_start:0.0001:l_c;
for i=1:length(x_c3)
    y_c3(i)=(rt+r_curve) - sqrt((r_curve^2)-(x_c3(i)-l_c)^2);
end
plot(((x_c3-l_c)/r_t),y_c3/r_t)
```

Aero 535

Jorns

Fall 2022

```
hold on
```

```
x3 = x_c3(1) - ((r_comb-y_c3(1))/tand(45));
plot((([(x3-l_c) (x_c3(1)-l_c)]/r_t), ([r_comb y_c3(1)]/r_t)
hold on
```

```
x5 = 0;
plot((([(x5-l_c) (x3-l_c)]/r_t), ([r_comb r_comb])/r_t)

% Setup system of equations
x_e1 = L - (r_c * sind(theta_i(3)));
y_e1 = r_e - r_t - (r_c*(1-cosd(theta_i(3))));
syms P Q S T
eqn1 = Q + sqrt(T) == 0;
eqn2 = (P*x_e1) + Q + sqrt((S*x_e1) + T) == y_e1;
eqn3 = P + ((1/2) * (S/sqrt(T))) == tand(theta_i(3));
eqn4 = P + ((1/2) * (S/sqrt((S*x_e1) + T))) == tand(theta_f(3));
% Solve the system of equations to find the coefficients
sol = solve([eqn1, eqn2, eqn3, eqn4], [P, Q, S, T]);
PSol = sol.P;
P = double(PSol);
QSol = sol.Q;
Q = double(QSol);
SSol = sol.S;
S = double(SSol);
TSol = sol.T;
T = double(TSol)
% Now solve for the coordinates of the parabola
x1 = 0:0.001:x_e1;
for i = 1:length(x1)
    y1(i) = (P*x1(i)) + Q + sqrt((S*x1(i)) + T);
end
% Move to account for the curvature offset
x_nozzle = x1+x_c(end);
y_nozzle = y1+y_c(end);
% Plot expansion part of the nozzle
plot ((x_nozzle/r_t), (y_nozzle/r_t))
axis equal
xlim([-2 14])
```

Aero 535

Jorns

Fall 2022

```
ylim([0 6])  
xlabel('Length of Nozzle (m)/r_t')  
ylabel('Radius of Nozzle (m)/r_t')  
title('JMT-1 Engine')  
hold off
```

Bell Nozzle Design (JMT-2)

```
% We can now start the design of our nozzle  
  
% Rao's Method for Bell Nozzle Design  
  
% Common Values  
  
r_t = 0.1092; % [m] Throat Radius  
  
r_c = 0.4 * r_t % [m] Radius of Curvature  
  
Ae_At = 60; % Expansion Ratio  
  
r_e = r_t * sqrt(Ae_At) % Exit Plane Radius  
  
P_amb = 101325; % [Pa] Ambient pressure at sea-level  
  
P_vac = 0; % [Pa] Ambient pressure at vacuum  
  
A_t = pi * (r_t^2)  
  
A_t = A_t  
  
A_e = A_t * 60;  
  
j=0;
```

```
for i = 60:10:100  
    if i == 60  
        j=j+1;  
        theta_i(1,j) = 46.5; % Initial parabola angle  
        theta_f(1,j) = 13.5; % Final parabola angle  
        lambda(1,j) = 0.972; % Nozzle correction factor  
    elseif i == 70  
        j=j+1;  
        theta_i(1,j) = 44; % Initial parabola angle  
        theta_f(1,j) = 10; % Final parabola angle  
        lambda(1,j) = 0.98; % Nozzle correction factor  
    elseif i == 80  
        j=j+1;  
        theta_i(1,j) = 40.5; % Initial parabola angle  
        theta_f(1,j) = 8; % Final parabola angle  
        lambda(1,j) = 0.9875; % Nozzle correction factor  
    elseif i == 90
```

Aero 535

Jorns

Fall 2022

```
j=j+1;

theta_i(1,j) = 38.5; % Initial parabola angle
theta_f(1,j) = 6; % Final parabola angle
lambda(1,j) = 0.99125; % Nozzle correction factor

else
    j=j+1;
    theta_i(1,j) = 36.5; % Initial parabola angle
    theta_f(1,j) = 4.5; % Final parabola angle
    lambda(1,j) = 0.9925;% Nozzle correction factor
end

% JMT-1 Shifting Equilibrium Values
T2_slse = T2se_sl .* lambda
T2_vacse = T2se_vac .* lambda
% JMT-1 Frozen Flow Values
T2_slfz = T2fz_sl .* lambda
T2_vacfz = T2fz_vac .* lambda

% Assume actual values lie 75% of way up from frozen flow boundary to
% shifting equilibrium boundary and get final values for thrust
delta_T2sl = T2_slse - T2_slfz
delta_Isp2sl = Isp2se_sl - Isp2fz_sl
T2_sl = T2_slfz + (0.75 .* delta_T2sl)
Isp2_sl = Isp2fz_sl + (0.75 * delta_Isp2sl)
delta_T2vac = T2_vacse - T2_vacfz
delta_Isp2vac = Isp2se_vac - Isp2fz_vac
T2_vac = T2_vacfz + (0.75 .* delta_T2vac)
Isp2_vac = Isp2fz_vac + (0.75 * delta_Isp2vac)

i = 60:10:100
figure
a20 = plot(i,T2_sl);
hold on
a21 = plot(i,T2_vac);
hold off
title('Thrust vs. Normalized Length Percentage')
ylabel('Thrust (kN)')
xlabel('Normalized Length (%)')
```

Aero 535

Jorns

Fall 2022

```
legend ([a20,a21],["Sea-Level Thrust","Vacuum Thrust"],'FontSize',6)

% Picking a normalized length of 80% re-calculate final performance metrics
% of the JMT-1 Engine

T2sl = T2_s1(3)

T2vac = T2_vac(3)

Isp2sl = Isp2_sl

Isp2vac = Isp2_vac

m_dot_2sl = T2sl / (Isp2sl * g0) * 1000

m_dot_2vac = T2vac / (Isp2vac * g0) * 1000

r_e = sqrt(A_e/pi)

d_e = r_e * 2

% Plot the expansion region of the nozzle

x1b = r_c * sind(theta_i(3)) % [m] Initial x-coordinate of expansion region of nozzle
y1b = r_t + (r_c * (1-cosd(theta_i(3)))) % [m] Initial y-coordinate of expansion region of nozzle

r_e = r_t * sqrt(Ae_At) % [m] Radius of the exit plane

L_15 = ((r_t*(sqrt(Ae_At)-1)) + (r_c*((1/cosd(15))-1))) / (tand(15)) % [m] Length of nozzle before shortening

L = 0.8 * L_15 % [m] Length of normalized length nozzle

T_L = T2sl/L % [N/m] Thrust to Length Ratio

% Plot the initial radius of curvature part

x_c2 = 0:0.0001:x1b;
for i = 1:length(x_c2)
    y_c2(i) = (r_t + r_c) - sqrt(((r_c)^2) - ((x_c2(i))^2));
end
plot ((x_c2/r_t),(y_c2/r_t))
hold on

l_c = 0.76/2.5;

theta = 45;

Ac = 2.5*At;

rt = sqrt(At/pi);

r_curve = 1.25*rt;

r_comb = sqrt(Ac/pi);

x_start = (l_c - r_curve*(sind(theta)));

x_c4 = x_start:0.0001:l_c;
for i=1:length(x_c4)
    y_c4(i)=(rt+r_curve) - sqrt((r_curve^2)-(x_c4(i)-l_c)^2);

```

```

end

plot((x_c4-l_c)/r_t),y_c4/r_t)

hold on

x4 = x_c4(1) - ((r_comb-y_c4(1))/tand(45));
plot(([([x4-l_c) (x_c4(1)-l_c)]/r_t),([r_comb y_c4(1)]/r_t)
hold on

x3 = 0;
plot(([([x3-l_c) (x4-l_c)]/r_t),([r_comb r_comb])/r_t)

% Setup system of equations

x_e1 = L - (r_c * sind(theta_i(3)));
y_e1 = r_e - r_t - (r_c*(1-cosd(theta_i(3))));
syms P Q S T

eqn1 = Q + sqrt(T) == 0;
eqn2 = (P*x_e1) + Q + sqrt((S*x_e1) + T) == y_e1;
eqn3 = P + ((1/2) * (S/sqrt(T))) == tand(theta_i(3));
eqn4 = P + ((1/2) * (S/sqrt((S*x_e1) + T))) == tand(theta_f(3));

% Solve the system of equations to find the coefficients

sol = solve([eqn1, eqn2, eqn3, eqn4], [P, Q, S, T]);
PSol = sol.P;
P = double(PSol)
QSol = sol.Q;
Q = double(QSol)
SSol = sol.S;
S = double(SSol)
TSol = sol.T;
T = double(TSol)

% Now solve for the coordinates of the parabola
x2 = 0:0.001:x_e1;
for i = 1:length(x2)
    y2(i) = (P*x2(i)) + Q + sqrt((S*x2(i)) + T);
end

% Move to account for the curvature offset
x_nozzle = x2+x_c2(end);
y_nozzle = y2+y_c2(end);

% Plot expansion part of the nozzle
plot ((x_nozzle/r_t),(y_nozzle/r_t))

```

Aero 535

Jorns

Fall 2022

```
axis equal  
xlim([-3 21])  
ylim([0 10])  
xlabel('Length of Nozzle (m)/r_t')  
ylabel('Radius of Nozzle (m)/r_t')  
title('JMT-2 Engine')  
hold off
```

Rocket Sizing Code:

Launch Vehicle Configuration (2-Stage)

```
clc; clear;  
  
g0 = 9.81; % [m/s^2]  
  
Isp_1 = 289; % [s]  
  
Isp_2 = 395; % [s]  
  
e1 = 0.05;  
  
e2 = 0.09;  
  
delta_v = 9300; % [m/s]  
  
  
% delta_v = (Isp_1 * g0 * log(R1)) + (Isp_2 * g0 * log(R2))  
  
  
syms alpha  
  
eqn1 = (Isp_1 * g0 * log((alpha * Isp_1 * g0)+1)/(alpha * Isp_1 * g0 * e1)) + (Isp_2 * g0 * log((alpha * Isp_2 * g0)+1)/(alpha * Isp_2 * g0 * e2))) == delta_v;  
  
alpha = vpasolve(eqn1, alpha);  
  
  
R1 = ((alpha * Isp_1 * g0)+1)/(alpha * Isp_1 * g0 * e1);  
R2 = ((alpha * Isp_2 * g0)+1)/(alpha * Isp_2 * g0 * e2);  
  
  
lambda1 = (1 - (e1*R1)) / (R1 - 1);  
lambda2 = (1 - (e2*R2)) / (R2 - 1);  
  
  
opt_mass_ratio = ((lambda1 + 1) / lambda1) * ((lambda2 + 1) / lambda2);  
  
  
mL = 120000; % [kg] Payload Mass  
m01 = opt_mass_ratio * mL; % [kg] Total Mass Including Payload  
  
  
m_02 = mL * ((1 + lambda2) / lambda2);  
m_01 = m_02 * ((1 + lambda1) / lambda1);
```

Aero 535

Jorns

Fall 2022

```
m2 = m_02 - mL;  
m1 = m_01 - m_02;
```

```
mb1 = m_01/R1;  
mb2 = m_02/R2;
```

Rocket Sizing (2-Stage)

```
clc; clear;  
g0 = 9.81;  
T_W1 = 1.5;  
T_W2 = 0.8;  
m_01 = 2418772.43;  
m_02 = 677149.40;
```

```
T1 = (T_W1 * g0 * m_01)/1000;  
T2 = (T_W2 * g0 * m_02)/1000;
```

```
T_JMT1 = 6913; % [N] Thrust at sea-level  
T_JMT2 = 1011.3; % [N] Thrust at vacuum  
num_eng1 = T1 / T_JMT1;  
num_eng2 = T2 / T_JMT2;  
T_act1 = 6 * T_JMT1;  
T_act2 = 6 * T_JMT2;
```

```
Isp_JMT1 = 279.9; % [s] Isp at sea-level  
Isp_JMT2 = 295.4; % [s] Isp at vacuum  
m_dot_1act = T_act1 / (Isp_JMT1 * g0) * 1000;  
m_dot_2act = T_act2 / (Isp_JMT2 * g0) * 1000;
```

```
m_b1 = 918088.11;  
m_b2 = 180943.45;  
m_p1 = m_01 - m_b1;  
m_p2 = m_02 - m_b2;  
t_b1 = m_p1 / m_dot_1act;  
t_b2 = m_p2 / m_dot_2act;
```

Aero 535

Jorns

Fall 2022

Launch Vehicle Configuration (3-Stage)

```
clc; clear;

g0 = 9.81; % [m/s^2]

Isp_1 = 289; % [s]

Isp_2 = 390; % [s]

Isp_3 = 395;

e1 = 0.05;

e2 = 0.07;

e3 = 0.19;

delta_v = 9500; % [m/s]

% delta_v = (Isp_1 * g0 * log(R1)) + (Isp_2 * g0 * log(R2))

syms alpha

eqn1 = (Isp_1 * g0 * log((alpha * Isp_1 * g0)+1)/(alpha * Isp_1 * g0 * e1)) + (Isp_2 * g0 * log((alpha * Isp_2 * g0)+1)/(alpha * Isp_2 * g0 * e2)) + (Isp_3 * g0 * log((alpha * Isp_3 * g0)+1)/(alpha * Isp_3 * g0 * e3)) == delta_v;

alpha = vpasolve(eqn1, alpha)

R1 = ((alpha * Isp_1 * g0)+1)/(alpha * Isp_1 * g0 * e1)

R2 = ((alpha * Isp_2 * g0)+1)/(alpha * Isp_2 * g0 * e2)

R3 = ((alpha * Isp_3 * g0)+1)/(alpha * Isp_3 * g0 * e3)

lambda1 = (1 - (e1*R1)) / (R1 - 1)

lambda2 = (1 - (e2*R2)) / (R2 - 1)

lambda3 = (1 - (e3*R3)) / (R3 - 1)

opt_mass_ratio = ((lambda1 + 1) / lambda1) * ((lambda2 + 1) / lambda2) * ((lambda3 + 1) / lambda3)

mL = 120000; % [kg] Payload Mass

m01 = opt_mass_ratio * mL % [kg] Total Mass Including Payload

m_03 = mL * ((1 + lambda3) / lambda3)

m_02 = m_03 * ((1 + lambda2) / lambda2)

m_01 = m_02 * ((1 + lambda1) / lambda1)

m3 = m_03 - mL

m2 = m_02 - m_03
```

Aero 535

Jorns

Fall 2022

```
m1 = m_01 - m_02
```

```
mb1 = m_01/R1  
mb2 = m_02/R2  
mb3 = m_03/R3
```

Rocket Sizing (3-Stage)

```
clc;  
clear;  
g0 = 9.81;  
T_W1 = 1.2;  
T_W2 = 0.7;  
T_W3 = 0.5;  
m_01 = 2825540.51;  
m_02 = 1596022.78;  
m_03 = 252788.91;
```

```
T1 = (T_W1 * g0 * m_01)/1000  
T2 = (T_W2 * g0 * m_02)/1000  
T3 = (T_W3 * g0 * m_03)/1000
```

```
T_JMT1 = 6913; % [N] Thrust at sea-level  
T_JMT2 = 1011.3; % [N] Thrust at vacuum  
num_eng1 = T1 / T_JMT1  
num_eng2 = T2 / T_JMT2  
num_eng3 = T3 / T_JMT2  
T_act1 = 5 * T_JMT1  
T_act2 = 11 * T_JMT2  
T_act3 = 2 * T_JMT2
```

```
Isp_JMT1 = 279.9; % [s] Isp at sea-level  
Isp_JMT2 = 295.4; % [s] Isp at vacuum  
m_dot_1act = T_act1 / (Isp_JMT1 * g0) * 1000  
m_dot_2act = T_act2 / (Isp_JMT2 * g0) * 1000  
m_dot_3act = T_act3 / (Isp_JMT2 * g0) * 1000
```

```
m_b1 = 1608005.7270020185076918395370261;  
m_b2 = 340933.99693052155398918675696194;
```

Aero 535

Jorns

Fall 2022

```
m_b3 = 144615.87310254638997976755377669;
m_p1 = m_01 - m_b1;
m_p2 = m_02 - m_b2;
m_p3 = m_03 - m_b3;
t_b1 = m_p1 / m_dot_1act;
t_b2 = m_p2 / m_dot_2act;
t_b3 = m_p3 / m_dot_3act;
```

Ascent Code:

```
% AEROSP-535

% Jacob Dewey, Tim Galmiche, Maria Mejia

% Final Project

clc, clear all, close all

%% Rocket Constants:

% Other Constants and Variables:

re = 6378000; % Radius of Earth [m]
g0 = 9.8; % Gravity of earth [m/s^2]
mu = 398600; %

ifpa = pi/2; % Initial Flight Angle [rad]
dt = 0.01; % Timestep [s]
targetAlt = 400e3; % Target Altitude [m]
altMargin = 500; % Margin for Altitude [m]
t_turn = 25; % Time of Gravity Turn [s]
init_dfpa = deg2rad(0.5); % Initial Gravity Turn Maneuver [rad]
orbSpeed = 7600;
speedMargin = 30;
fpaMargin = 15;

check = 0;
check1 = 0;
check2 = 0;

%% Simulation:

% pitchAngleVec = deg2rad(linspace(20,35,1000));
pitchAngleVec = deg2rad(34);
pitchAngleVec2 = deg2rad(-8);
pitchj = 1;
for pitchi = 1:length(pitchAngleVec)
```

Aero 535

Jorns

Fall 2022

```
pitchAngle = pitchAngleVec(pitchi);

pitchAngle2 = pitchAngleVec2(pitchj);

% Stage 1:

s1.mi = 2873102.09; % Intial Stage Mass [kg]

s1.tf = 79.1; % Burn time [s]

s1.thrust = 34565000; % Stage Thrust [N]

s1.mdot = 12588; % Mass Flow Rate of Engines [kg/s]

s1.stage = 1;

% Stage 2:

s2.mi = 1825995.31; % Intial Stage Mass [kg]

s2.tf = 396.5; % Burn time [s]

s2.thrust = 18203400; % Stage Thrust [N]

s2.mdot = 3650.08; % Mass Flow Rate of Engines [kg/s]

s2.stage = 2;

% Stage 3:

s3.mi = 270157.33; % Intial Stage Mass [kg]

s3.tf = 233.3; % Burn time [s]

s3.thrust = 2022600; % Stage Thrust [N]

s3.mdot = 521.44; % Mass Flow Rate of Engine [kg/s]

s3.stage = 3;

massEmpty = s3.mi - s3.mdot*s3.tf;

% Intial Timestep:

m = s1.mi - s1.mdot*dt;

a = s1.thrust/m - g0*sin(ifpa);

v = dt*a;

alt = v*dt*sin(ifpa);

xdis = v*dt*cos(ifpa);

fpa = ifpa;

theta = fpa;

i = 2;

for t = dt:dt:s1.tf+s2.tf+s3.tf

    if t < s1.tf-.00001 % First Stage Burn

        [m(i), a(i), v(i), alt(i), xdis(i), fpa(i), fpadot(i)] = sim(m(i-1), v(i-1),
        alt(i-1), xdis(i-1), fpa(i-1), fpadot(i-1), pitchAngle2, dt, s1, t, t_turn, init_dfpa);

        if check1 == 0 && alt(i-1) > 40e3
```

```

check1 = 1;
pitch1 = i;
pitch1t = t;

elseif check2 == 0 && alt(i) > 200e3

    check2 = 1;
    pitch2 = i;
    pitch2t = t;

end

i = i+1;

elseif abs(t-s1.tf) < 0.00001 % Staging Step from Stage 1 to Stage 2

    fprintf('Staging from 1 to 2: t + %1.2f seconds\n',t)

meco = i-1;

m(i) = s2.mi - s2.mdot*dt;

a(i) = s2.thrust/m(i)*cos(fpa(i-1)-fpa(i-1)) - g0*sin(fpa(i-1));

v(i) = v(i-1) + dt*a(i);

alt(i) = alt(i-1) + v(i)*dt*sin(fpa(i-1));

xdis(i) = xdis(i-1) + v(i)*dt*cos(fpa(i-1));

r = re + alt(i);

fpadot(i) = -(g0/v(i) - v(i)/r)*cos(fpa(i-1));

fpa(i) = fpa(i-1) + dt*fpadot(i);

gravityLoss(i) = g0*sin(fpa(i))*dt;

if check1 == 0 && alt(i-1) > 40e3

    check1 = 1;
    pitch1 = i;
    pitch1t = t;

elseif check2 == 0 && alt(i) > 200e3

    check2 = 1;
    pitch2 = i;
    pitch2t = t;

end

i = i+1;

elseif t < s1.tf+s2.tf-.00001 % Second Stage Burn

    [m(i), a(i), v(i), alt(i), xdis(i), fpa(i), fpadot(i)] = sim(m(i-1), v(i-1),
alt(i-1), xdis(i-1), fpa(i-1), pitchAngle, pitchAngle2, dt, s2, t, t_turn, init_dfpfa);

    gravityLoss(i) = g0*sin(fpa(i))*dt;

```

```

if check1 == 0 && alt(i-1) > 40e3
    check1 = 1;
    pitch1 = i;
    pitch1t = t;

elseif check2 == 0 && alt(i) > 200e3
    check2 = 1;
    pitch2 = i;
    pitch2t = t;

end

i = i+1;

elseif abs(t-s1.tf-s2.tf) < 0.00001 % Staging Step from Stage 2 to Stage 3
    seco = i-1;
    fprintf('Staging from 2 to 3: t + %1.2f seconds\n',t)

if alt(i-1) >= 200e3
    pitchAngles3 = pitchAngle2;
else
    pitchAngles3 = pitchAngle;
end

m(i) = s3.mi - s3.mdot*dt;
a(i) = s3.thrust/m(i)*cos(pitchAngle-fpa(i-1)) - g0*sind(fpa(i-1));
v(i) = v(i-1) + dt*a(i);
alt(i) = alt(i-1) + v(i)*dt*sin(fpa(i-1));
xdis(i) = xdis(i-1) + v(i)*dt*cos(fpa(i-1));
r = re + alt(i);
fpadot(i) = -(g0/v(i) - v(i)/r)*cos(fpa(i-1)) +
s3.thrust/(v(i)*m(i))*sin(pitchAngles3-fpa(i-1));
fpa(i) = fpa(i-1) + dt*fpadot(i);
gravityLoss(i) = g0*sin(fpa(i))*dt;

if check1 == 0 && alt(i-1) > 40e3
    check1 = 1;
    pitch1 = i;
    pitch1t = t;

elseif check2 == 0 && alt(i) > 200e3
    check2 = 1;
    pitch2 = i;

```

Aero 535

Jorns

Fall 2022

```
pitch2t = t;
end

i = i+1;

elseif t < s1.tf+s2.tf+s3.tf-.00001 % Third Stage Burn
[m(i), a(i), v(i), alt(i), xdis(i), fpa(i), fpadot(i)] = sim(m(i-1), v(i-1),
alt(i-1), xdis(i-1), fpa(i-1), pitchAngle, pitchAngle2, dt, s3, t, t_turn, init_dfpa);
gravityLoss(i) = g0*sin(fpa(i))*dt;

if check1 == 0 && alt(i-1) > 40e3
    check1 = 1;
    pitch1 = i;
    pitch1t = t;
elseif check2 == 0 && alt(i) > 200e3
    check2 = 1;
    pitch2 = i;
    pitch2t = t;
end

i = i+1;

if abs(v(i-1)-orbSpeed) < speedMargin && abs(alt(i-1)-targetAlt) < altMargin &&
abs(rad2deg(fpa(i-1))) < fpaMargin
    teco = i-1;
    fprintf('Orbital Insertion Complete :) at t+%1.2f seconds\n\n',t)
    fprintf('Final Altitude: %1.3f km\n',alt(i-1)/1000)
    fprintf('Final Velocity: %1.3f km/s\n',v(i-1)/1000)
    fprintf('Final Mass: %1.2f kg \n',m(i-1))
    fprintf('Turn Time t+ %1.2f s\n',t_turn)
    fprintf('Turn Angle = %1.2f deg\n',rad2deg(init_dfpa))
    fprintf('Pitch Angle 1 = %1.2f deg at t+%1.2f
seconds\n',rad2deg(pitchAngle),pitch1t)
    fprintf('Pitch Angle 2 = %1.2f deg at t+%1.2f
seconds\n',rad2deg(pitchAngle2),pitch2t)
    check = 1;
    break
elseif alt(i-1) < 0
    fprintf('Rocket Crashed :( at t+%1.2f seconds\n\n',t)
```

```

        break

    elseif abs(v(i-1)-orbSpeed) < speedMargin*0.1

        fprintf('Orbital Speed reached at t+%1.2f seconds\n',t)

        fprintf('Altitude = %1.2f km\n\n',alt(i-1)/1000)

        break

    elseif t > s1.tf+s2.tf+s3.tf-.00001

        fprintf('No Orbit, Ran Out of Fuel :( at t+%1.2f seconds\n',t)

        fprintf('Alt = %1.2f km, V = %1.2f km/s\n\n', alt(i-1)/1000, v(i-1)/1000)

        break

    end

end

if check == 1

    break

end

% figure;

% plot(x,rad2deg(finalFPA))

% ylabel('Final FPA [deg]')

%

% figure;

% plot(x,finalVel/1000)

% ylabel('Final Vel [km/s]')

%

% figure;

% plot(x,finalAlt/1000)

% ylabel('Final Alt[km]')

%% Plots:

tl = length(alt);

tv = linspace(0,tl*dt,tl);

mecotxt = '\leftarrow MECO';

secotxt = '\leftarrow SECO';

tecotxt = 'Third Engine Cutoff, Orbit Achieved \rightarrow';

pitch1txt = '\leftarrow Pitch Manuever 1';

pitch2txt = '\leftarrow Pitch Manuever 2';

```

Aero 535

Jorns

Fall 2022

```
% Altitude vs Time Plot:
```

```
figure;

plot(tv,alt/1000,'LineWidth',2)

text(tv(meco),alt(meco)/1000,mecotxt)

text(tv(seco),alt(seco)/1000,secotxt)

text(tv(i-1),alt(i-1)/1000,tecotxt,'HorizontalAlignment','right')

text(tv(pitch1),alt(pitch1)/1000,pitch1txt)

text(tv(pitch2),alt(pitch2)/1000,pitch2txt)

title('Altitude vs Time')

xlabel('Time [sec]')

ylabel('Altitude [km]')

print('AltvsTime','-dpng','-r300')
```

```
% Mass vs Time Plot:
```

```
figure;

plot(tv,m,'LineWidth',2)

title('Mass vs Time')

xlabel('Time [sec]')

ylabel('Mass [kg]')

print('MassvsTime','-dpng','-r300')
```

```
% Altitude vs Downrange Distance Plot:
```

```
figure;

hold on

plot(xdis/1000,alt/1000,'LineWidth',2)

text(xdis(meco)/1000,alt(meco)/1000,mecotxt)

text(xdis(seco)/1000,alt(seco)/1000,secotxt)

text(xdis(i-1)/1000,alt(i-1)/1000,tecotxt,'HorizontalAlignment','right')

text(xdis(pitch1)/1000,alt(pitch1)/1000,pitch1txt)

text(xdis(pitch2)/1000,alt(pitch2)/1000,pitch2txt)

title('Altitude vs Downrange Distance')

xlabel('Downrange Distance [km]')

ylabel('Altitude [km]')

print('AltvsXdis','-dpng','-r300')
```

```
% Acceleration vs Time Plot:
```

```
figure;

plot(tv,a,'LineWidth',2)
```

Aero 535

Jorns

Fall 2022

```
title('Acceleration vs Time')

xlabel('Time [sec]')

ylabel('Acceleration [m/s^2]')

print('AccvsTime',' -dpng', '-r300')

% G-Loading vs Altitude Plot:

figure;

plot(alt/1000,a/g0,'LineWidth',2)

text(alt(meco)/1000,a(meco)/g0,mecotxt)

text(alt(seco)/1000,a(seco)/g0,secotxt)

text(alt(i-1)/1000,a(i-1)/g0,tecotxt,'HorizontalAlignment','right')

text(alt(pitch1)/1000,a(pitch1)/g0,pitch1txt)

text(alt(pitch2)/1000,a(pitch2)/g0,pitch2txt)

title('G-Loading vs Altitude')

xlabel('Altitude [km]')

ylabel('G-Loading')

print('GvsTime',' -dpng', '-r300')

% Velocity vs Time Plot:

figure;

plot(tv,v,'LineWidth',2)

title('Velocity vs Time')

xlabel('Time [sec]')

ylabel('Velocity [m/s]')

print('VelvsTime',' -dpng', '-r300')

% FPA vs Time Plot:

figure;

plot(tv,rad2deg(fpa),'LineWidth',2)

title('Flight Path Angle vs Time')

xlabel('Time [sec]')

ylabel('Flight Path Angle [deg]')

print('FPAvsTime',' -dpng', '-r300')

%

% figure;

% plot(tv(20.51/dt:t1),rad2deg(fpadot(20.51/dt:t1)),'LineWidth',2)

% title('dfpa vs Time')

% xlabel('Time [sec]')
```

Aero 535

Jorns

Fall 2022

```
% ylabel('dfpa [deg/s]')

%% Functions:

function [m, a, v, alt, xdis, fpa, fpadot] = sim(mIn, vIn, altIn, xdisIn, fpaIn, thetaIn,
thetaIn2, dt, stage, t, kickTime, kickInit)

if t < kickTime

    re = 6378000;

    g0 = 9.8; % Gravity of earth

    m = mIn - stage.mdot*dt;

    a = (stage.thrust/m) - g0*sin(fpaIn);

    v = vIn + a*dt;

    alt = altIn + v*sin(fpaIn)*dt;

    xdis = xdisIn + v*cos(fpaIn)*dt;

    r = re+alt;

    fpadot = 0;

    fpa = fpaIn;

elseif abs(t-kickTime) < 0.000001

    re = 6378000;

    g0 = 9.8; % Gravity of earth

    m = mIn - stage.mdot*dt;

    a = (stage.thrust/m) - g0*sin(fpaIn);

    v = vIn + a*dt;

    alt = altIn + v*sin(fpaIn)*dt;

    xdis = xdisIn + v*cos(fpaIn)*dt;

    r = re+alt;

    fpadot = -kickInit;

    fpa = fpaIn - kickInit;

elseif altIn < 40e3

    re = 6378000;

    g0 = 9.8; % Gravity of earth

    m = mIn - stage.mdot*dt;

    a = (stage.thrust/m) - g0*sin(fpaIn);

    v = vIn + a*dt;

    alt = altIn + v*sin(fpaIn)*dt;

    xdis = xdisIn + v*cos(fpaIn)*dt;

    r = re+alt;
```

Aero 535

Jorns

Fall 2022

```
fpadot = - (g0/v - v/(r))*cos(fpaIn);

fpa = fpaIn + dt*fpadot;

elseif altIn >= 40e3 && altIn < 200e3

re = 6378000;

g0 = 9.8; % Gravity of earth

m = mIn - stage.mdot*dt;

a = (stage.thrust/m*cos(thetaIn-fpaIn)) - g0*sin(fpaIn);

v = vIn + a*dt;

alt = altIn + v*sin(fpaIn)*dt;

xdis = xdisIn + v*cos(fpaIn)*dt;

r = re+alt;

fpadot = - (g0/v - v/r)*cos(fpaIn) + stage.thrust/(v*m)*sin(thetaIn-fpaIn);

fpa = fpaIn + dt*fpadot;

elseif altIn >= 200e3

re = 6378000;

g0 = 9.8; % Gravity of earth

m = mIn - stage.mdot*dt;

a = (stage.thrust/m*cos(thetaIn-fpaIn)) - g0*sin(fpaIn);

v = vIn + a*dt;

alt = altIn + v*sin(fpaIn)*dt;

xdis = xdisIn + v*cos(fpaIn)*dt;

r = re+alt;

fpadot = - (g0/v - v/r)*cos(fpaIn) + stage.thrust/(v*m)*sin(thetaIn2-fpaIn);

fpa = fpaIn + dt*fpadot;

end

end
```

Sizing Code:

```
% AEROSP-535

% Jacob Dewey, Tim Galmiche, Maria Mejia

% Final Project

clc, clear all, close all

s1.mi = 2873102.09; % Intial Stage Mass [kg]

s1.tf = 79.02; % Burn time [s]

s1.thrust = 34565000; % Stage Thrust [N]
```

Aero 535

Jorns

Fall 2022

```
s1.mdot = 12588; % Mass Flow Rate of Engines [kg/s]
s1.stage = 1;

% Stage 2:

s2.mi = 1825995.31; % Intial Stage Mass [kg]
s2.tf = 346.86; % Burn time [s]
s2.thrust = 16180800; % Stage Thrust [N]
s2.mdot = 4171.52; % Mass Flow Rate of Engines [kg/s]
s2.stage = 2;

% Stage 3:

s3.mi = 270157.33; % Intial Stage Mass [kg]
s3.tf = 233.25; % Burn time [s]
s3.thrust = 2022600; % Stage Thrust [N]
s3.mdot = 521.44; % Mass Flow Rate of Engine [kg/s]
s3.stage = 3;

massEmpty = s3.mi - s3.mdot*s3.tf;
dt = 0.01;

m = s1.mi - s1.mdot*dt;
a = s1.thrust/m;
v = dt*a;
i = 2;

for t = dt:dt:(s3.tf+s2.tf+s1.tf)

    if t < s1.tf-.00001 % First Stage Burn
        m(i) = m(i-1) - s1.mdot*dt;
        a(i) = s1.thrust/m(i);
        v(i) = v(i-1) + dt*a(i);
        i = i+1;
    elseif abs(t-s1.tf) < 0.00001 % Staging Step from Stage 1 to Stage 2
        fprintf('Staging from 1 to 2: t + %1.2f seconds\n',t)
        m(i) = s2.mi - s2.mdot*dt;
        a(i) = s2.thrust/m(i);
        v(i) = v(i-1) + dt*a(i);
        i = i+1;
    elseif t < s1.tf+s2.tf-.00001 % Second Stage Burn
        m(i) = m(i-1) - s2.mdot*dt;
        a(i) = s2.thrust/m(i);
```

Aero 535

Jorns

Fall 2022

```
v(i) = v(i-1) + dt*a(i);

i = i+1;

elseif abs(t-s1.tf-s2.tf) < 0.00001 % Staging Step from Stage 2 to Stage 3

    fprintf('Staging from 2 to 3: t + %1.2f seconds\n',t)

m(i) = s3.mi - s3.mdot*dt;

a(i) = s3.thrust/m(i);

v(i) = v(i-1) + dt*a(i);

i = i+1;

elseif t < s1.tf+s2.tf+s3.tf-.00001 % Third Stage Burn

    m(i) = m(i-1) - s3.mdot*dt;

    a(i) = s3.thrust/m(i);

    v(i) = v(i-1) + dt*a(i);

    i = i+1;

end

end

tvec = linspace(dt,length(m)*dt,length(m));

figure;
plot(tvec,m)

figure;
plot(tvec,a)

figure;
plot(tvec,v)
```