Assignment #1: Mission Analysis

Daniel Card*, Torrence Gue†, and Marc Chattrabhuti‡
University of Michigan, Ann Arbor, MI, 48109

Due: Friday, September 25 at 5:00 P.M. ET

I. Nomenclature

Δv	=	Change in velocity increment (m/s)
I_{sp}	=	Specific impulse (s)
T	=	Thrust (N)
$m_{0,i}$	=	Initial rocket mass of i^{th} stage (kg)
$m_{f,i}$	=	Final rocket mass of i^{th} stage (kg)
m_i	=	Mass of structural and propellant for individual i^{th} stage (kg)
$m_{p,i}$	=	Mass of propellant for i^{th} manevuar (kg)
g_0	=	Gravitational acceleration at sea level (m/s^2)
m_l	=	Payload Mass
R_i	=	Mass ratio of the i^{th} stage
α	=	Lagrange Multiplier constant
λ_i	=	payload ratio of i^{th} stage
TWR_i	=	Thrust to weight ratio
σ_i	=	Structural coefficient of the i^{th} stage
α	=	Lagrange multiplier
λ_i	=	Payload fraction of the i^{th} stage
G	=	Universal gravitational constant $(m^3 kg^{-1} s^{-2})$
ψ	=	Flight angle (rad)
ϕ	=	Rockets azimuthal angle (rad)
r	=	Rockets radius (km)

II. Introduction

In the early 1960s, U.S. President John Kennedy set the ambitious goal of landing people on the moon by the end of the decade. This was an incredible undertaking given that at the time the U.S. had only sent a handful of astronauts into low earth orbit. In response to this challenge, NASA identified three proposed concepts (shown in Fig. 1) for the mission: earth orbit rendezvous (EOR), lunar orbit rendezvous (LOR), and direct ascent (DA). In the DA architecture, the entire spacecraft lands on the moon and then launches again for return to the earth. This was initially the leading candidate favored by NASA as it is the simplest, requiring only one dedicated vehicle. However, this single spacecraft necessarily must have a large amount of mass and structure to be able to safely land and re-launch from the moon. Sending this type of payload from orbit would in turn require the development of an enormous launch vehicle.

The EOR architecture employs multiple launches of smaller spacecraft components which are then assembled in earth's orbit. The assembled vehicle transits to the moon where it lands, and after completion of operations, launches again for return to earth. While ultimately requiring smaller launch vehicles than the DA, the EOR's major weakness stems from its reliance on both the ability to launch multiple payloads successfully and in turn to assemble a functional vehicle in

*Graduate Aerospace Engineering, AIAA Student Member, Aerospace Honors Society.
†Undergraduate Aerospace Engineering, AIAA Student Member, Aerospace Honors Society, External Vice President.
‡Graduate Aerospace Engineering, Aerospace Honors Society

space. In the early 1960s, there have never been a demonstrated rendezvous of two space vehicles in orbit—let alone the capability to assemble a crewed vehicle.

The final concept, the LOR, requires one launch vehicle that has two components: a lunar lander and a mother ship. Both ships are launched together from a single launch vehicle and enter lunar orbit in tandem. The lander then detaches and descends to the moon. After completion of the mission, the lander ascends for rendezvous with the mother ship. The astronauts transfer from the lander to the mother ship, jettison the lander, and return to earth. This approach, famously championed by John Houbolt, requires a lower overall mass than the EOR and DA since less propellant is required to land and launch the small lunar lander. Similarly, as the lander is abandoned in lunar orbit, the propellant required to return the mother ship to earth is also smaller. As with the EOR concept, however, the disadvantage of LOR is that in the early 1960s, no vehicles had ever performed a rendezvous in space.

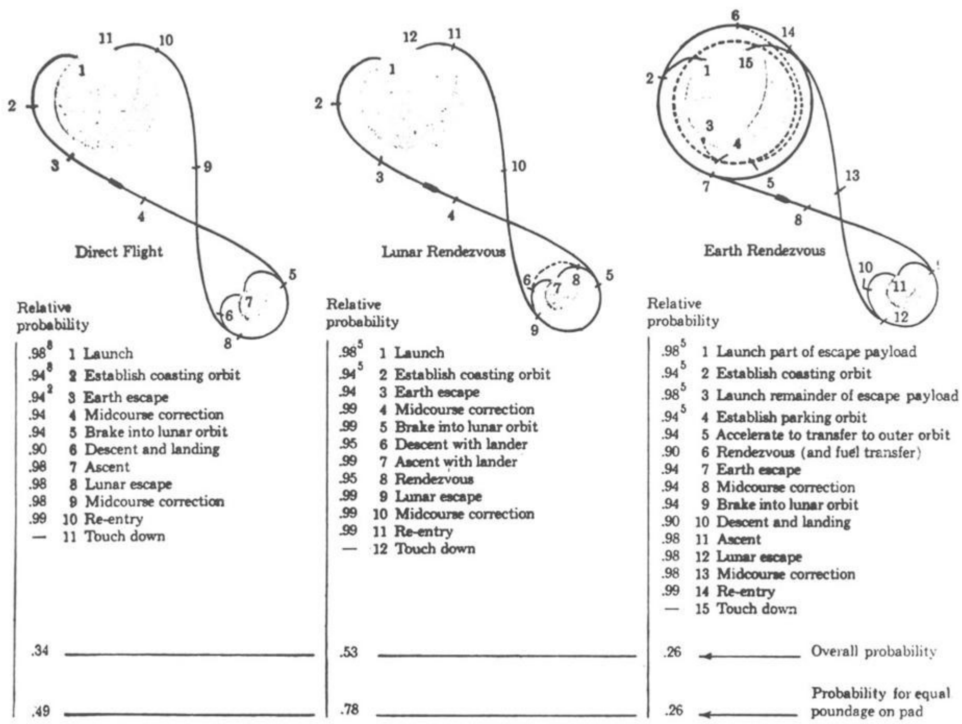

Fig. 1 Three proposed architectures for reaching the moon. Image credit: NASA history office.

The debate over the which architecture to followed led to several intense discussions at NASA. At one point, President Kennedy even had to step in during a press conference to smooth over a heated argument between administrators. Ultimately, however, it was decided that the technical challenges with DA and EOR were too great to be overcome in a decade. This left lunar orbit rendezvous as the down selected architecture, and the Apollo program with the Saturn V launch rocket is the result. The reader can find a nice summary of the story here.

In this report, we put ourselves in the role of NASA engineers in the early 1960s trying to understand the trade between the direct ascent and a lunar orbit return architecture. We explicitly calculate (subject to some simplifying assumptions) the mass budget for the mission and the required size of the launch vehicles. We then offer a preliminary design for the launch vehicle and simulate its trajectory, illustrating key features in the launch profile.

III. Key Assumptions for Analysis

We outline in the following the key assumptions we use in our trade study comparing the DA and LOR. This includes a discussion of the vehicle dry masses, the delta-v budget, and the propulsion characteristics.

A. Vehicle Dry Masses

The LOR has two vehicles for performing in-space maneuvers, a lunar lander and a command and service module (CSM). The DA architecture in contrast has only one vehicle, the CSM. We show in Table 1 the estimated dry masses for these two vehicles. These estimates are based on the required structural mass to support human life for the required mission duration as well as the necessary mass for avionics, propellant storage, and consumables.

Table 1 Dry masses for the CSM and lunar lander.

Vehicle	Dry Mass, kg
Lunar Lander	4280
Command and Service Module	11900

B. Delta-V Budget

We show in Table 2 the delta-v requirements for each aspect of the two mission architectures as depicted in Fig. 1. The major difference is that the lunar lander detaches from the CSM for the descent to the moon. This vehicle thus performs the descent and ascent maneuvers. The lander is then jettisoned before the CSM returns to earth. We note that the launch to LEO delta-v requirement is lower than a nominal 7.7 km/s. This stems from the fact that we assume the vehicle will launch due East from Cape Canaveral, thereby starting out with an initial velocity (from the earth's rotation) with respect to the center of the earth of 0.4 km/s.

Table 2 Delta-V requirements for both mission architectures.

Maneuver	Delta-V, $\frac{km}{s}$	Vehicle for DA	Vehicle for LOR
Launch to LEO and Establish Coasting Orbit	7.3	Launch Vehicle	Launch Vehicle
Earth Escape	3.2	3rd Stage of Launch Vehicle	3rd Stage of Launch Vehicle
Midcourse Corrections	≈ 0	N/A	N/A
Brake into Lunar Orbit	0.9	CSM	CSM
Descent and Landing	2.5	CSM	Lunar Lander
Ascent	2.2	CSM	Lunar Lander
Rendezvous	0.1	N/A	Lunar Lander
Lunar Escape	0.9	CSM	CSM

C. Average Properties or Propulsion Systems Used for Maneuvering

We show in Table 3 the expected specific impulse for the different stages of the mission. For the in-space maneuvers performed by either the CSM or lunar lander, we have assumed a propellant based on N_2O_4 oxidizer and Aerozine-50 fuel. This choice of propellant balances the need for high fuel economy and the need for a high reliability. As a bipropellant (fuel and oxidizer), this mixture has a higher specific impulse than the common alternative used for in-space propulsion, hydrazine-based monopropellants. The other advantage of this propellant is that it is hypergolic. It will spontaneously ignite when the oxidizer and fuel are combined. This eliminates the need for an ignition system, thereby greatly enhancing reliability (if the ignitor failed when trying to launch off the moon, the astronauts would be stranded!).

For the launch vehicle stages, we baseline a kerosene first stage (LOX/RP-1) and liquid hydrogen (LOX/LH2) upper stages. Kerosene propellant has a lower upperbound in achievable specific impulse compared to liquid hydrogen. It therefore has a lower fuel economy, which translates to more propellant required for a given maneuver. Indeed, for a delta-v ranging from 0 – 12 km/s, a rocket with RP1 will require on average 1.35 times more mass of propellant than an engine with liquid hydrogen. However, the LOX/RP-1 mixture has a density that is 3 times higher than the LOX/LH2 mixture. As a result, a launch vehicle using LOX/RP-1 will require approximately 4 times less volume for propellant

Table 3 Assumed average properties of the vehicles.

Propellant	Vehicle	Average Specific Impulse, s	Structural Coefficient, ϵ	Thrust to Weight at Ignition
N_2O_4/Aerozine-50	CSM and Lunar Lander	311	N/A	N/A
LOX/RP-1	1st Stage	283	0.05	1.2
LOX/LH2	2nd Stage	311	0.07	0.7
LOX/LH2	2nd or 3rd Stage	421	0.19	0.5

storage than a vehicle with LOX/LH2. This is a critical consideration for the lower stage which spends most of its time climbing out of the atmosphere. The increased volume of the rocket (and by extension) cross-sectional are becomes a prohibitively large source drag for a LOX/LH2 first stage. Once above atmosphere where the smaller, upper stages are ignited, drag profile is no longer a concern. The higher performance LOX/LH2 is therefore the choice for these upper stages.

The estimates for specific impulse in Table 3 for this initial trade study represent average values expected for each stage. For example, state of the art RP-1 based rockets (see Table 4) have specific impulses that range from 263 s at sea level to 304 s at vacuum conditions. As we expect this first stage to climb through the atmosphere to reach the barrier of space, we use a value that represents the average of the two environmental extremes, 283 s. For the second stage with its LOX/LH2 rocket, which may start in atmosphere and then continue to climb into space, we similarly use, 311 s, an average value between the sea level performance (200 s) and vacuum performance (421 s) For the third, LH2 stage, which ignites when the rocket is in space, we assume the vacuum value of specific of 421 s.

We include in Table 3 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. As noted above, since RP-1 is a denser propellant, it requires less volume for storage. The storage tanks consequently are smaller allowing for lower structural coefficient. Conversely, 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.

Finally, we show in Table 3 the requirements for the thrust to weight of each vehicle stage. For the first stage, the initial trajectory is vertical. The rocket therefore must provide sufficient acceleration (thrust to weight greater than 1) to overcome the full force of gravity. In subsequent stages as the rocket trajectory gains altitude and shifts its trajectory to horizontal with respect to the surface of the earth, the requirements for acceleration relax. The lower thrusts have the additional benefits of lowering the effective g-loading on the astronauts.

D. Baselined Engines for Trajectory Analysis

We show in Table 4 the properties of the two real engines that were under development at the beginning of the Apollo program, the RP-1 based F-1 engine and the LH2 based J-2. The F-1 interestingly was developed before the 1960s—before there were even designs for a launch vehicle. The J-2 was developed in parallel with the initial formulation of the Apollo program. While we use the average performance metrics in Table 3 to evaluate our initial design, we return to the properties shown in Table 4 to estimate the number of engines and total thrust levels required for each stage.

Table 4 Properties of the engines considered for the rocket stages.

Engine	Propellant	Sea level thrust, kN	Vacuum thrust, kN	SL I_{sp}, s	Vacuum I_{sp}, s	Mass of Engine, kg	Engine Diameter, m
F1	LOX/RP-1	6770	7770	263	304	8400	3.7
J-2	LOX/LH2	486.2	1033	200	421	1788	2.1

IV. Analysis

We present in the following our design and sizing for both the payload (CSM and Lunar Lander wet masses) as well as the launch vehicle.

A. Payload Mass

We calculate here the payload mass of the launch vehicle for both the DA and LOR architectures. This estimate includes the total wet mass, i.e. propellant and dry contributions, of the vehicle(s) that will be sent on a translunar injection to the moon. These estimates are based on using the delta-v budget shown in Table 1, dry masses indicated in Table 2, and the performance metrics for the Aerozine-50 propulsion system shown in Table 3.

1. Direct Ascent, [5 pts]

For a DA architecture, we know the total delta-v required during the lunar phase. This phase starts at the lunar braking maneuver and ends at the lunar escape. To calculate the payload mass, we use the rocket equation Eq. 1 below:

$$m_0 = m_f e^{\Delta v/g_0 I_{sp}} \tag{1}$$

where m_0 is the payload mass, m_f is the dry CSM mass at lunar escape, Δv is the total delta-v for the lunar phase, g_0 is the gravitational constant of 9.8 $\frac{m}{s^2}$, and I_{sp} is the specific impulse of the Aerozine-50 propulsion system. Since we know that these maneuvers all use the same propulsion system in space, we can plug in the sum of the delta-v's for the maneuvers of the lunar phase for Δv. As found in our previous tables, for DA we know that m_f is 11900 kg, Δv is equal to 6.5 km/s, and the I_{sp} is 311 seconds. Solving for the payload mass, we find that the Payload Mass for the DA architecture is 100408 kg.

2. Lunar Orbit Rendezvous, [10 pts]

For a LOR architecture, we must use the rocket equation again, however we cannot use the sum of the delta-v contributions. This is because during the lunar phase, the lunar lander separates from the CSM, and later recombines with it. Because of this, we must use the rocket equation separate for each time propellant is burned which will give us a system of three equations and three unknowns. These unknowns are is the propellant burned during the lunar brake, moon landing/ascent, and lunar escape $m_{p,\text{brake}}$, $m_{p,\text{moon}}$, and $m_{p,\text{escape}}$ respectively. For each equation, we must consider what the initial and final masses for each will be. We can start from the escape maneuver, as we know the final mass will simply be the combined dry mass of the CSM and the lander. We then know that the initial mass will equal the final mass plus the fuel burned during the maneuver. For the moon landing/ascent stage we know that the final mass will be the dry mass of the lunar lander, and the initial mass will be that lander mass plus the propellant burned. Finally, we know that the final mass for lunar brake will be the combined dry mass of the CSM and lander, as well as their required propellant for the escape and moon maneuvers ($m_{p,\text{moon}}$, and $m_{p,\text{escape}}$). The initial mass will then be this initial mass, plus the propellant mass burned during the brake. We now three rocket equations in terms of $m_{p,\text{brake}}$, $m_{p,\text{moon}}$, and $m_{p,\text{escape}}$, which we will solve for. Following the form of equation 1 above, the system of equations is listed below in Equations 2, 3, and 4.

$$m_{CSM} + m_{lander} + m_{p,\text{escape}} = (m_{CSM} + m_{lander})e^{\Delta v_{escape}/g_0 I_{sp}} \tag{2}$$

$$m_{lander} + m_{p,\text{escape}} + m_{p,\text{moon}} = (m_{lander} + m_{p,\text{escape}})e^{\Delta v_{moon}/g_0 I_{sp}} \tag{3}$$

$$m_{CSM} + m_{lander} + m_{p,\text{escape}} + m_{p,\text{moon}} + m_{p,\text{brake}} = (m_{CSM} + m_{lander} + m_{p,\text{escape}} + m_{p,\text{moon}})e^{\Delta v_{brake}/g_0 I_{sp}} \tag{4}$$

where m_{CSM} is dry mass of the CSM, m_{lander} is the dry mass of the lunar lander, Δv is the effective velocity increment for each maneuver, g_0 is the gravitational constant of 9.8 $\frac{m}{s^2}$, and I_{sp} is the specific impulse of the Aerozine-50 propulsion system in a vacuum. These values are all given to us in Tables 1, 2 and 3. Solving for $m_{p,\text{brake}}$, $m_{p,\text{moon}}$, and $m_{p,\text{escape}}$ using MATLAB we find that the values are 12594kg, 16393kg, and 4088 kg, respectively. Solving for the launch vehicle payload mass, which is simply the weight of the initial mass of the spacecraft before the braking maneuver, we find that it is equal to 49256 kg.

The calculated payload masses for both DA and LOR architectures can be found below in Table 5. The payload mass for the lunar orbit rendezvous payload mass is noticeably smaller than that of the direct ascent payload mass.

Table 5 Estimated payload masses for DA and LOR architectures.

Arhitecture	Payload Mass, kg
Direct Ascent	100408 kg
Lunar Orbit Rendezvous	49256 kg

B. Launch Vehicle Configuration

We baseline a three-stage launch vehicle for propelling the payloads shown in Table 5 into cislunar space. This choice represents a compromise between the gains that result from multi-staging and the complexity involved with increasing the number of stages.

1. Mass of Each Stage, [20 pts]

The optimal configuration of each stage is determined by the minimization of the inverse of the payload ratio m_{01}/m_L. Assuming that we are given Δv, ϵ, and I_{sp} we can minimize Eq. 5:

$$\frac{m_{01}}{m_L} = \prod_{i=1}^{n} 1 + \frac{(R_i - 1)}{1 - \epsilon_i R_i} \tag{5}$$

Given the summation of Δv in each stage of the rocket. We can modify the rocket equation into a constraint in which we modify mass ratio of each stage R_i in Eq. 6:

$$G(R_i) = \Delta v - \sum_{i=1}^{N} I_{sp,i}\; g_0 \ln(R_i) = 0 \tag{6}$$

In order to minimize the inverse payload ratio given our modified rocket equation constraint, we must use the method of Lagrange Multipliers on parameter R_i for the first stage to the nth stage of the rocket. Therefore Eq. 7 must be satisfied:

$$\frac{\partial}{\partial R_i} \ln\left(\frac{m_{01}}{m_{0L}}\right) + \alpha \frac{\partial}{\partial R_i} G(R_i) \tag{7}$$

Thus, R_i becomes a function of the Lagrange multiplier α shown in Eq. 8:

$$R_i = \frac{\alpha I_{sp_i} g_0 + 1}{\alpha I_{sp_{(i)}} g_0 \epsilon_i} \tag{8}$$

Substituting in the mass fraction R_i into the modified rocket equation that serves as our constraint, we obtain Eq. 9:

$$\Delta v = \sum_{i=1}^{N} I_{sp,i}\; g_0 \ln\left(\frac{\alpha I_{sp,i}\; g_0 + 1}{\alpha I_{sp,i}\; g_0 \epsilon_i}\right) \tag{9}$$

Solving for the Lagrange Multiplier with the I_{sp}, ϵ_i, and Δv from Table 2 we calculate that $\alpha = -4.3 * 10^{-4}$. In addition to solving for mass fraction R_i we can also solve for the payload fraction λ using Eq. 10:

$$\lambda_i = \frac{1 - R_i \epsilon_i}{R_i - 1} \tag{10}$$

Solving for the payload ratio of each stage, we then calculate the wet mass of each stage using Eq. 11:

$$m_{0,i} = m_{i+1}\left(\frac{1+\lambda_i}{\lambda_i}\right) \tag{11}$$

The mass difference between each of these stages determines the mass of each stage based on Eq. 12:

$$m_{0,i} = m_{0,i} - m_{0,i+1} \tag{12}$$

where $m_{0,i}$ is the wet mass of stage i. The masses of each stage for both the Direct Ascent and the Lunar Orbit Rendezvous were calculated and tabulated in the following table:

Table 6 Mass of each stage for two architectures

Architecture	1st Stage mass, ($\times 10^6\ kg$)	2nd Stage mass, ($\times 10^6\ kg$)	3rd Stage mass, ($\times 10^6\ kg$)	Total Mass Including Payload, ($\times 10^6\ kg$)
DA	3.8285	1.0644	0.23360	5.2269
LOR	1.8781	0.52215	0.11459	2.5641

The mass of every Direct Ascent stage is around twice that of the Lunar Orbit Rendezvous, with much more propellant and structural mass required to get the required payload into orbit.

2. Rocket Sizing, [20 pts]

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 Table 3, the actual engine performance shown in Table 4, and the masses reported in Table 6.

For the thrust level at ignition (as reported in Table 3), we assume the first stage F-1 rockets will be at sea level and the upper stage J-2 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 architecture by determining the optimal circular diameter that would fit the indicated number of engines.

To find the thrust at ignition T_i of any given stage i, we can simply take the thrust-to-weight ratio TWR_i and multiply it by the initial stage mass m_{0i} and the gravity constant g_0. We can use this for each stage and for both architectures. This equation becomes:

$$T_i = TWR_i\ m_{0,i}g_0 \tag{13}$$

We can then find the minimum number of engines at each stage necessary to provide thrust since we know the maximum thrust capabilities for the F-1 and J-2 engines, T_{engine}. For both architectures, during stage 1 we use F-1 engines and for stages 2 and 3 we use J-2 engines. We can use the following equation to find the number of engines required at each stage, which is simply the required initial thrust at each stage divided by the maximum thrust of a single engine.

$$\text{Number of Engines} = \frac{T_i}{T_{engine}} \tag{14}$$

We round up this number to find the minimum number of engines needed. To find fuel flow rate of each stage $\dot{m}_i$ we can use the equation relating fuel flow rate the i^{th} stage $I_{sp,i}$ and thrust T_i:

$$\dot{m}_i = \frac{T_i}{g_0\, I_{sp,i}} \tag{15}$$

Next, we need to find the amount of propellant in each stage. This can be found using the structural coefficient at each stage ϵ_i since we know the mass of each stage. Since we calculated the stage mass m_i earlier, we can use it to find the structural mass of each stage m_{si}:

$$m_{si} = \epsilon_i m_i \tag{16}$$

We also know that m_i is equal to the sum of m_{si} and m_{pi}, the propellant mass of the i^{th} stage. Thus we have Eq. 17:

$$m_{pi} = m_i - m_{si} \tag{17}$$

Then, since we know the stage propellant mass and the mass flow rate, we can calculate the burn time of each stage:

$$\text{Burn Time} = \frac{m_{pi}}{\dot{m}_i} \tag{18}$$

To find the stage diameter, we must use the calculations for circle packing inside a circle, seen in Appendix [B]. For 10 engines, the packing will be such that the diameter of the stage is approximately equal to 3.813 times the diameter of one of the engines. For 5 engines, in optimum packing the diameter of the stage will be equal to 3 times the diameter of one of the engines. Finally, for 1 engines the stage diameter will simply be the diameter of the engine itself.

Table 7 Engine characteristics for each architecture.

Architecture	Stage	Thrust at Ignition, kN	Flow Rate, $\frac{kg}{s}$	Burn Time, s	Engine Type	Number of Engines	Stage Diameter, m
DA	1st	67700	26267	138.5	F1	10	14.1
LOR	1st	33850	13133	141.3	F1	5	9.99
DA	2nd	10330	2504	395.4	J2	10	8.01
LOR	2nd	5165	1252	403.4	J2	5	5.67
DA	3rd	2066	500.8	377.9	J2	2	4.20
LOR	3rd	1033	250.4	385.6	J2	1	2.10

Comparing the two architectures, we can see here that the required thrust for the first stage for DA is much higher than that of LOR. We can also see that most values of the first stage are much greater than those of LOR. This makes sense, because for DA architecture the mass of each stage is almost twice that of the stage masses for LOR, and as a result the requirements for thrust are much greater. As a result, the number of required engines for DA is also higher for stages 1 and 2, and is only the same for stage 3 because of the capabilities of the J-2 engine. Of course, as a result the stage diameter for stages 1 and 2 are greater for DA than LOR. For all three stages, DA will also require higher fuel flow rates (around twice as much for each stage).

For burn time, it is interesting to note that for all stages it is the same for both DA and LOR architecture. Overall however, we see that the DA architecture demands much higher performance characteristics. A DA rocket will be heavier, require more thrust, has more engines and a higher fuel flow rate for the same mission (landing on the moon).

3. Engine Down-Select, [5 pts]

As we can see from the previous calculations, DA architecture, while much simpler in execution than LOR, requires much higher performance characteristics. DA architecture would require twice as many engines for stages one and two, as well as much more fuel and structural mass. As a result, we can imagine how much more expensive as well as technically complex a DA rocket would be compared to an LOR rocket. Even though a mid-orbit rendezvous may be difficult, the economical and technological disadvantages of getting a massive launch vehicle into orbit would outweigh this concern. Thus, it is logical to choose LOR as the superior architecture for a moon landing. In Table 8 and 9 below, we compared our LOR architecture to the actual Saturn V rocket.

Table 8 Mass of each stage for Saturn V and LOR

Architecture	1st Stage mass, ($\times 10^6$ kg)	2nd Stage mass, ($\times 10^6$ kg)	3rd Stage mass, ($\times 10^6$ kg)	Total Mass Including Payload, ($\times 10^6$ kg)
Saturn V	2.2900	.49620	.12300	5.2269
LOR	1.8781	0.52215	0.11459	2.5641

Table 9 Engine characteristics for Saturn V and LOR architecture.

Architecture	Stage	Thrust at Ignition, kN	Flow Rate, $\frac{kg}{s}$	Burn Time, s	Engine Type	Number of Engines	Stage Diameter, m
Saturn V	1st	35100	13618	168	F1	5	10.1
LOR	1st	33850	13133	141.3	F1	5	9.99
Saturn V	2nd	5141	1246	360	J2	5	10.1
LOR	2nd	5165	1252	403.4	J2	5	5.67
Saturn V	3rd	1033	250.4	500	J2	1	6.6
LOR	3rd	1033	250.4	385.6	J2	1	2.10

Comparing our calculations LOR architecture to the actual Saturn V rocket, we find that the specifications vary somewhat. The most important note is that the Saturn V's architecture is different than ours as its third stage does two separate burns at different times, rather than one continuous burn. As a result, we can see that the mass distribution for each stage is different, and for the Saturn V the second stage burns much shorter while the overall burn of the third stage is much longer. The wet masses of the Saturn V rocket were 2290000 kg for the first stage, 1093900 kg for the second, and 123000 kg for the third, while our rocket masses were 2564114 kg, 685995 kg, and 163850 kg respectively. The reason the Saturn V first stage weighs so much more than ours is mostly likely due to the fact that we did not include propellant masses for maneuvering thrusters in our estimates, and we assumed that all propellant mass will be used in the primary thrusters. The listed first and second stage thrusts of the Saturn V rocket are also different than those that we calculated, which could be due to the fact that they were measured at different conditions or our given engine specifications are off.

One of the biggest contributors to our calculated rocket mass is the fact that our assumed performance characteristics (such as I_{sp}, thrust to weight, etc.) are simply estimates, and may not be completely accurate with the specifications of the actual F1 and J2 engines on the Saturn V. This reason is also what causes variations in our mass flow rate and burn times. We disregarded the changing I_{sp} of rocket engines as they went up in altitude, instead choosing an "average" I_{sp} value to simplify our calculations. In reality, the performance of the engines are constantly changing as the rocket goes up into the atmosphere, meaning that the mass flow rate will change as time goes on. As a result, this can make our calculated stage masses, mass flow rate, and burn times.

Finally, we can see that despite the number of engines for each stage being the same, the diameters for the second and third stages are different. It makes sense that the Saturn V's second and third stage diameters are larger than our values, as there must be some space between the rockets and the fairings that house them. For the first stage, we expect the stage diameters to be the same as the rockets are not housed inside a fairing.

C. Simulated Rocket Launch, [30 pts]

We simulate in this section a 2D launch profile for our rocket. For simplicity, we have assumed

- The initial part of the trajectory is the result of a gravity turn
- After the gravity turn is complete, we gimbal the rocket thrust vector to provide a constant pitch angle, θ_{gimb}, with respect to the local horizon
- We neglect the rotation of the earth.
- We neglect atmospheric drag and lift and solve for the trajectory in 2D
- 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.
- We target a destination altitude of 200 km by adjusting the gimbal angle of the thruster.

The launch concept of operations is shown in Table 10. We note we solve the trajectory up until (7.3 km/s) at an altitude of 200km. This leaves sufficient propellant mass remaining on board to perform the last orbital insertion maneuver (3.2 km/s). The third stage engine cutoffs at this point and will-re-ignite later for the escape trajectory. The model for atmospheric pressure for this calculation is shown in the Appendix.

Table 10 Concept of operations for launch profile of rocket.

Mission Time, s	Event
0	Vertical Liftoff
Δ 12	Pitching Maneuver 1° from vertical. Rocket starts gravity turn
1st Stage burn time	Main engine cut out (MECO) and separation
Δ 0	2nd Stage Ignition
Δ 0	Thrusters angled to pitch at θ_{gimb} with respect to local horizon
Δ 2nd Stage burn time	Main engine cut out (MECO) and separation
Δ 0	3rd Stage Ignition
3rd Stage cutoff	Velocity reaches 7.3 $\frac{km}{s}$

Firstly in order to simulate the rockets launch and trajectory, we will formulate the governing equations that will determine the path that the rocket will partake through its launch, gravity turn, and pitch of the rocket engine. The first formulation will be the change in flight angle shown below in Eq. 19, with Eq. 20 being the alternative when the rocket starts pitching:

$$\dot{\psi} = \left(\frac{v}{r} - \frac{g_0}{r}\right)\cos\psi \tag{19}$$

$$\dot{\psi} = \left(\frac{v}{r} - \frac{g_0}{r}\right)\cos\psi + \frac{T}{vm}\sin(\theta - \psi) \tag{20}$$

Shown above in Equations 19, 20 the term $\left(\frac{v}{r} - \frac{g_0}{r}\right)\cos\psi$ arises from the changing angle and will evaluate to zero if the rocket is in a circular orbit (the flight path angle will be constant), then the term $\frac{T}{vm}\sin(\theta - \psi)$ results from the gimbaling angle of the rocket engine. Then the next governing equation will be the increase in speed, Eq. 21 with the alternative after the rocket starting pitching shown in Eq. 22 below:

$$\dot{v} = \frac{T}{m} - g_0 \sin\psi \tag{21}$$

$$\dot{v} = \frac{T}{m}\cos(\theta - \psi) - g_0 \sin\psi \tag{22}$$

Since the change in velocity is in the spacecrafts reference frame the term $g_0 \sin\psi$ gives the acceleration that the spacecraft experiences fighting gravity in its own frame. Then the term $\frac{T}{m}\cos(\theta - \psi)$ in Eq. 22 results from the rocket thrusting with the additional `cosine` term resulting from the pitch of the rocket.

Next is the change in the rockets azimuthal angle where Eq. 23 is the governing equation during a no-pitch launch, and Eq. 24 is the alternative expression for when the rocket starts pitching:

$$\dot{\phi} = \frac{v}{r}\cos\psi \tag{23}$$

$$\dot{\phi} = \frac{v}{r}\cos\psi + \frac{T}{vm}\sin(\theta - \psi) \tag{24}$$

Shown above in Eq. 23 the expression $\frac{v}{r}\cos\psi$ gives the change in azimuthal angle from differentiating arc length. The additional term $\frac{T}{vm}\sin(\theta - \psi)$ in Eq. 24 occurs from the pitch of the rocket engine. Then finally, Eq. 25 gives the increase in altitude and Eq. 26 gives the mass flow rate. Notable is that neither of these expressions differ from a vertical launch to a pitching maneuver. Thus their expressions are:

$$\dot{r} = v\sin\psi \tag{25}$$

$$\frac{dm}{dt} = -\dot{m} \tag{26}$$

Using these expressions from above and putting into matrix form gives Eq. 27 below:

$$\begin{bmatrix} \frac{dm}{dt} \\ \frac{dv}{dt} \\ \frac{dr}{dt} \\ \frac{d\psi}{dt} \\ \frac{d\phi}{dt} \end{bmatrix} = \begin{bmatrix} -\dot{m} \\ \frac{T}{m}\underbrace{\cos(\theta - \psi)}_{\text{Pitching Maneuver}} - g_0\sin\psi \\ v\sin\psi \\ \left(\frac{v}{r} - \frac{g_0}{r}\right)\cos\psi + \underbrace{\frac{T}{vm}\sin(\theta - \psi)}_{\text{Pitching Maneuver}} \\ \frac{v}{r}\cos\psi + \underbrace{\frac{T}{vm}\sin(\theta - \psi)}_{\text{Pitching Maneuver}} \end{bmatrix} \tag{27}$$

Using the systems of Equations in the matrix given from Eq. 27 and using Matlab's `ode45` integrating function we will numerically integrate the systems of equations to approximate the rockets trajectory. However, to determine the gimbaling angle θ_{gimb} we will approximate it to end at the target altitude $h = 200\ km$ at a velocity of $v = 7.3\ \frac{km}{s}$. Looking to Listing 4 in which we run the entire rocket launch and then find when the rocket reaches 7.3 $\frac{km}{s}$ and then determine the gimbal angle θ_{gimb} that will give us the target altitude. Doing this numerical approximation by iterating our gimbal angle we ultimately found that the gimbal angle needed was $\theta_{gimb} = 25.95°$ to meet these conditions.

Simulating the g-loading will be done by using the thrust values throughout the launch and written as Eq. 28:

$$g_{\text{load}} = \frac{T}{g_0 m} \tag{28}$$

Where T is the thrust from the rocket, m is the mass at a given time, and then g_0 is the gravitational acceleration at sea level. This expression shown above in Eq. 28 will give the resultant g's that the rocket will experience throughout the rockets flight and stage separations.

Using the Matlab code found in the Appendix in Listing 4 and using thrust values to determine g-loading, the rockets trajectory, its downrange position and accounting for stage separations gives the following Fig. 2 below.

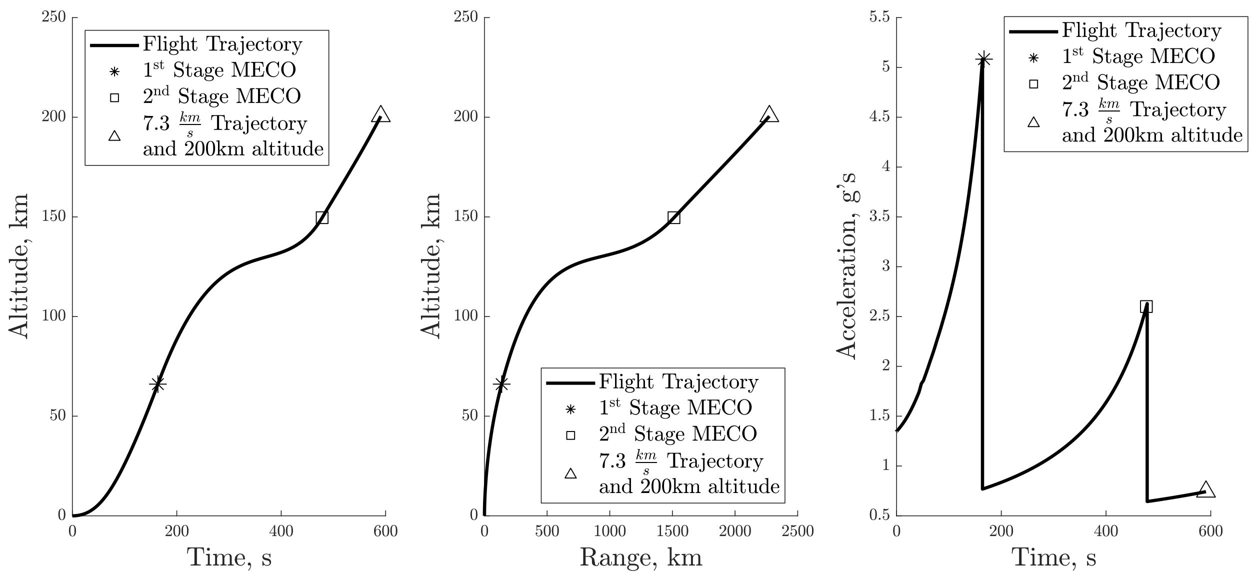

Fig. 2 Characteristics of calculated launch profile. The different events from Table 8 are also labeled.

Looking above to Fig. 2 we can see the entire launch profile of ascent, and several stage separations. The far-left subplot shows the altitude of the rocket and how it varies over time. The middle subplot shows the altitude and the downrange location of the rocket. Then finally, the far-right subplot shows the accelerations that the rocket experiences with every "dip" in the plot indicating stage separation as per the legend.

V. Discussion and Conclusion, [5 pts]

From our analysis, we found the performance characteristics of a theoretical launch vehicle capable of delivering a payload to and from the Moon using two separate architectures, Direct Ascent and Lunar Orbit Rendezvous. First, we used the rocket equation and our given delta-v requirements to calculate the require payloads that each launch vehicle would have to deliver to the moon. Next, we used required payload mass to find each stage's optimal mass by minimizing the inverse payload fraction given the vehicle's structural coefficients, stage specific impulse, and Δv_{stage}. We were then able to determine the engine characteristics including stage thrust to weight ratio, number of engines per stage, and propellant mass flow rate for each architecture. From our calculations, we found that while both concepts are capable of carrying a CSM or CSM/lander, the weight and performance specifications of the LOR architecture are much more viable than those of a DA rocket. This is due to the fact that a DA rocket would require more than twice the mass and number of engines than that of the LOR rocket in order to complete the same mission. While the LOR rocket might require more complicated maneuvers to achieve a lunar rendezvous, the economical and manufacturing advantages of being nearly half the size of a DA rocket outweighs this concern.

Using our numerically approximated values, we were able to simulate a 2D launch profile for our theoretical LOR launch vehicle, which would reach a target orbit of 200 km above the surface of the Earth. Looking to Fig. 2, the 1st stage spends most of its energy to direct the rocket mainly straight up which is made evident in the middle subplot where it only travels approximately 140km downrange whereas the 2nd stage travels approximately 1370 km downrange. This intuitively makes sense since the 1st stage's main purpose is to accelerate the launch vehicle out of the lower atmosphere to lower drag to allow the 2nd stage to begin to circularize the rockets trajectory and reach a target altitude.

The g-loading shows that most of the acceleration occurs during the last seconds of the 1st stages burn before MECO. This poses structural challenges since this will impose high stress on the launch vehicle increasing risk of failure. This risk can, and is typically mitigated by simply reducing the thrust from the engines to limit the experienced g's to a safe margin in which the launch vehicle has been engineered to withstand.

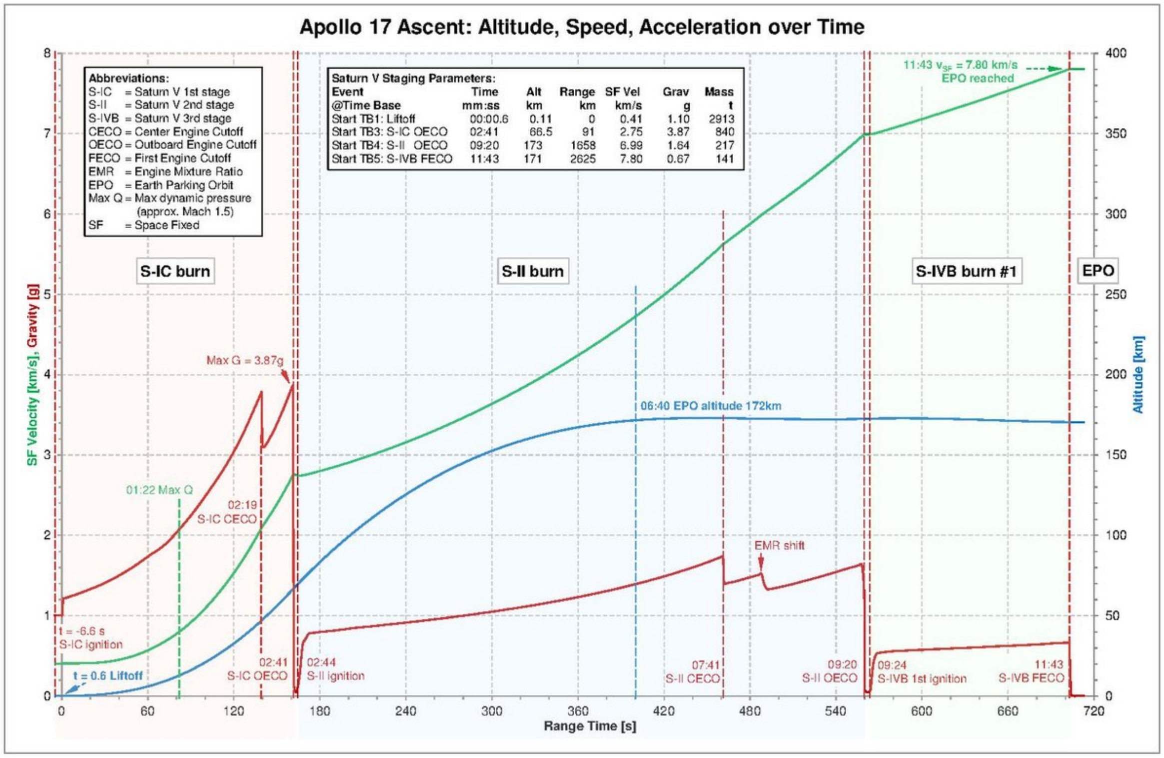

Fig. 3 Saturn V's actual g-loading profile.

When comparing the Saturn V launch profile, one of the significant differences that can be seen in Fig. 3 is the g loading of the space craft. The Saturn V controls thrust during Max Q by cutting off the center engine then outboard engines prior to the first stage separation. This way astronauts only experience 4 g's instead of 5 g's based off of our simulation. Additionally, our LOR vehicle also goes to an altitude of 200 km compared to the 173km of the Saturn V rocket.

Going further to improve our design, we could implement g-loading limitations that will lower the maximum acceleration that the rocket will experience to historical values typically ranging from 0 to 1.5g's. We could also implement lift and drag effects since neglecting these effects can impact the rockets trajectory. One last improvement we could implement is to calculate maximum dynamic pressure, the point in the trajectory in which the rocket experiences the highest stress due to atmospheric drag. We could modify our code to determine this maximum dynamic pressure and significantly reduce the thrust of the engines at this point in the trajectory, which would preserve structural integrity and ensure astronaut safety on our space vehicle.

Appendix

Table 11 U.S. standard atmospheric model.

GEO Potential Altitude above Sea Level, m	Temperature, $°C$	Acceration of Gravity, $\frac{m}{s^2}$	Absolute Pressure, $10^4\ \frac{N}{m^2}$	Density, $10^{-1}\ \frac{kg}{m^3}$	Dynamic Viscosity, $10^{-5}\ \frac{N \cdot s}{m^2}$
-1000	21.50	9.810	11.39	13.57	1.821
0	15.00	9.807	10.13	12.25	1.789
1000	8.50	9.804	8.988	11.12	1.758
2000	2.00	9.801	7.950	10.07	1.726
3000	-4.49	9.797	7.012	9.093	1.694
4000	-10.98	9.794	6.166	8.194	1.661
5000	-17.47	9.791	5.405	7.364	1.628
6000	-23.96	9.788	4.722	6.601	1.595
7000	-30.45	9.785	4.111	5.900	1.561
8000	-36.94	9.782	3.565	5.258	1.527
9000	-49.90	9.776	2.650	4.135	1.458
10000	-49.90	9.776	2.650	4.135	1.458
15000	-56.50	9.761	1.211	1.948	1.422
20000	-56.50	9.745	0.5529	0.8891	1.422
25000	-51.60	9.730	0.2549	0.4008	1.448
30000	-46.64	9.715	0.1197	0.1841	0.1475
40000	-22.80	9.684	0.0287	0.03996	1.601
50000	-2.5	9.654	0.007978	0.01027	1.704
60000	-26.13	9.624	0.002196	0.003097	1.584
70000	-53.57	9.594	0.00052	0.0008283	1.438
80000	-74.51	9.564	0.00011	0.0001846	1.321

Payload Mass Matlab Code

Listing 1 Matlab Code for determining payload mass.

```
%   ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
clear all; clc; close all
set(groot,'defaulttextinterpreter','latex');
set(groot, 'defaultAxesTickLabelInterpreter','latex');
set(groot, 'defaultLegendInterpreter','latex');
format long
%   ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
ispaerozine = 311;
deltavtotal = 17.1;
csm = 11900;
lander = 4280;
g0 = 9.8;
e = 2.7183;

%DA
deltavspace = (.9+2.2+2.5+.9)*1000; %sum of deltav for all post-lunar injection maneuvers
m0DA = csm/(e^(-(deltavspace)/(g0*ispaerozine))) %from rocket equation

% m0/mf = e(deltav/(goisp))
syms mp_brake mp_escape mp_moon
eqn1 = (csm+ lander+ mp_brake+ mp_escape+ mp_moon)/(csm+ lander+ mp_moon+ mp_escape) == e^((0.9e3)/(g0*ispaerozine)); % Brake into lunar orbite
eqn2 = (lander+ mp_moon)/(lander) == e^((2.5 + 2.2 + 0.1)*1e3/(g0*ispaerozine)); % Descent and landing
eqn3 = (csm+ mp_escape)/csm == e^((0.9e3)/(g0*ispaerozine)); % Lunar escape
sol = solve([eqn1,eqn2,eqn3],[mp_brake, mp_moon, mp_escape]);

m0LOR = csm + lander + double(sol.mp_brake + sol.mp_escape + sol.mp_moon)
```

Launch Vehicle Configuration Matlab Code

Listing 2 Matlab Code to determine the best launch vehicle configuration.

```
%   ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
clear all; clc; close all
set(groot,'defaulttextinterpreter','latex');
set(groot, 'defaultAxesTickLabelInterpreter','latex');
set(groot, 'defaultLegendInterpreter','latex');
%   ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Section B.1

%Solving for the Lagrange coefficient alpha in R_i optimizing equation
%Constants values:
g = 9.8;
Isp1 = 283; Isp2 = 311; Isp3 = 421; Isp_csm_lun = 311;
e1 = 0.05;e2 = 0.07;e3 = 0.19;
deltav_tot = 10500; %10.5 km/s
ml_da = 100408; %kg
ml_lor = 49256; %kg

a = sym('a');
% syms a
a_1 = (a.*Isp1.*g + 1)./(a .*Isp1.*g.*e1);
a_2 = (a.*Isp2.*g + 1)./(a .*Isp2.*g.*e2);
a_3 = (a.*Isp3.*g + 1)./(a .*Isp3.*g.*e3);
term1 = Isp1.*g.*log(a_1);
term2 = Isp2.*g.*log(a_2);
term3 = Isp3.*g.*log(a_3);
eqn1 = deltav_tot == term1 + term2 + term3 ;
ans_a = vpasolve(eqn1, a);

r1 = subs(a_1,ans_a); %payload ratios
r2 = subs(a_2,ans_a);
r3 = subs(a_3,ans_a);

lam1 = (1 - r1.*e1)./(r1-1);
lam2 = (1 - r2.*e2)./(r2-1);
lam3 = (1 - r3.*e3)./(r3-1);

%Direct ascent
m03 = ml_da.*((1+lam3)./(lam3));
m02 = m03.*((1+lam2)./(lam2));
m01 = m02.*((1+lam1)./(lam1));

mass3 = m03 - ml_da;
mass2 = m02 - m03;
mass1 = m01 - m02;

%LOR
m03_l = ml_lor.*((1+lam3)./lam3);
m02_l= m03_l.*((1+lam2)./(lam2));
m01_l = m02_l.*((1+lam1)./(lam1));

mass3_l = m03_l - ml_lor;
mass2_l = m02_l - m03_l;
mass1_l = m01_l - m02_l;
```

Listing 3 Matlab Code to determine the best launch vehicle configuration.

```
%   ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
clear all; clc; close all
set(groot,'defaulttextinterpreter','latex');
set(groot, 'defaultAxesTickLabelInterpreter','latex');
set(groot, 'defaultLegendInterpreter','latex');
%   ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Section B.1

%Solving for the Lagrange coefficient alpha in R_i optimizing equation
%Constants values:
g = 9.8;
Isp1 = 283; Isp2 = 311; Isp3 = 421; Isp_csm_lun = 311;
e1 = 0.05;e2 = 0.07;e3 = 0.19;
deltav_tot = 10500; %10.5 km/s
ml_da = 100408; %kg
ml_lor = 49256; %kg

a = sym('a');
% syms a
a_1 = (a.*Isp1.*g + 1)./(a .*Isp1.*g.*e1);
a_2 = (a.*Isp2.*g + 1)./(a .*Isp2.*g.*e2);
a_3 = (a.*Isp3.*g + 1)./(a .*Isp3.*g.*e3);
term1 = Isp1.*g.*log(a_1);
term2 = Isp2.*g.*log(a_2);
term3 = Isp3.*g.*log(a_3);
eqn1 = deltav_tot == term1 + term2 + term3 ;
ans_a = vpasolve(eqn1, a);

r1 = subs(a_1,ans_a); %payload ratios
r2 = subs(a_2,ans_a);
r3 = subs(a_3,ans_a);

lam1 = (1 - r1.*e1)./(r1-1);
lam2 = (1 - r2.*e2)./(r2-1);
lam3 = (1 - r3.*e3)./(r3-1);

%Direct ascent
m03 = ml_da.*((1+lam3)./(lam3));
m02 = m03.*((1+lam2)./(lam2));
m01 = m02.*((1+lam1)./(lam1));

mass3 = m03 - ml_da;
mass2 = m02 - m03;
mass1 = m01 - m02;

%LOR
m03_l = ml_lor.*((1+lam3)./lam3);
m02_l= m03_l.*((1+lam2)./(lam2));
m01_l = m02_l.*((1+lam1)./(lam1));

mass3_l = m03_l - ml_lor;
mass2_l = m02_l - m03_l;
mass1_l = m01_l - m02_l;
```

Simulating Launch

Listing 4 Matlab Code to simulate rocket launch.

```matlab
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
clear all; clc; close all
set(groot,'defaulttextinterpreter','latex');
set(groot, 'defaultAxesTickLabelInterpreter','latex');
set(groot, 'defaultLegendInterpreter','latex');
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
g0 = 9.8; % Gravity at sea level
Re = 6371e3;% Radius of Earth
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
m1 = 2564117.61; % Initial mass of stage 1
burn1 = 164.10;  % Burn duration of stage 1
mdot1 = 10872.58; % Mass flow rate of stage 1 kg/s

m2 = 685996.35;  % Initial mass of stage 2
burn2 = 314.49;  % Burn duration of stage 2
mdot2 = 1544.04; % Mass flow rate of stage 2 kg/s

m3 = 163850.07;  % Initial mass of stage 3
burn3 = 476.99;  % Burn duration of stage 3
mdot3 = 194.59;  % Mass flow rate of stage 3 kg/s

masses = [m1; m2; m3];
mdots = [mdot1; mdot2; mdot3];
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%%% Solver Format
% launchODE(time,y, parameters,mdot,Thrust_SL,Thrust_vac,Gimbal,verticalLaunch,fixedPitch,targetPitch)
% paramters : [Mass, radius, speed, flight-path-angle, true anomaly]

% First Stage launch - Vertical flight
opts = odeset('MaxStep',1);
param = [masses(1), Re, 0, deg2rad(90), 0];
[t1, x1] = ode45(@(t,y) launchODE(t,y, mdots(1), 5*6770e3, 5*7770e3, 0, true, false, 0),[0, 12], param, opts);

% First Stage launch - Gravity Turn
param = x1(end, :);
param(4) = deg2rad(90 - 1);
[t2, x2] = ode45(@(t,y) launchODE(t,y, mdots(1), 5*6770e3, 5*7770e3, 0, false, false, 0),[12, burn1], param, opts);
x1(end, :) = []; t1(end) = []; % Adjust the stop condition to determine the acceleration without discontinuities

% Second Stage - Fixed Pitch
gimbal_angle = deg2rad(25.95); % Fixed angle ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

param = x2(end, :);
param(1) = masses(2);
[t3, x3] = ode45(@(t,y) launchODE(t,y, mdots(2), 5*486.2e3, 5*1033.1e3, 0, false, true, gimbal_angle),[t2(end), t2(end) + burn2], param,opts);
x2(end, :) = []; t2(end) = []; % Adjust the stop condition to determine the acceleration without discontinuities

% Third Stage - Fixed Pitch
param = x3(end, :);
param(1) = masses(3);
[t4, x4] = ode45(@(t,y) launchODE(t,y, mdots(3), 486.2e3, 1033.1e3, 0,false, true, gimbal_angle),[t3(end), t3(end) + burn3], param,opts);
x3(end, :) = []; t3(end) = []; % Adjust the stop condition to determine the acceleration without discontinuities

% Collect and append all data
time = [t1; t2; t3; t4];
dat = [x1; x2; x3; x4];
accel = dlmread('thrust_vals/dat')./g0; % Read-in thrust values for g-loading
accel = accel(1:3588);
time_accel = dlmread('thrust_vals/time');% Read-in time values for g-loading
```

```matlab
time_accel = time_accel(1:max(size(accel)));

% Determine when the spacecraft reaches 7.3 km/s
i = 1; inloop = true;
while inloop
    if dat(i,3) > 7.3e3
        dat(i:end, :) = []; % Delete all other entries
        time(i:end) = [];
        inloop = false;
    end
    i = i + 1;
end
h_max = (dat(end,2)-Re)/10^3; % Used to determine height
%   ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Figures
figure()
subplot(1,3,1)
hold on
plot(time, (dat(:,2) - Re)./10^3, 'k', 'linewidth', 2)
scatter(t2(end), (x2(end,2) - Re)/10^3, 100, 'k*')
scatter(t3(end), (x3(end,2) - Re)/10^3, 100, 'ks')
scatter(time(end), (dat(end,2) - Re)./10^3, 100, 'k^')
xlabel('Time, s', 'fontsize', 16)
ylabel('Altitude, km', 'fontsize', 16)
legend({'Flight Trajectory','$1\textsuperscript{st}$ Stage MECO', '$2\textsuperscript{nd}$ Stage MECO', ['$7.3\ \frac{km}{s}$ Trajectory', newline, ' and 200km altitude']},'location', 'northwest', 'fontsize', 13)

subplot(1,3,2)
hold on
plot(Re.*dat(:,5)./10^3, (dat(:,2) - Re)./10^3, 'k', 'linewidth', 2)
scatter(Re*x2(end, 5)/10^3, (x2(end,2) - Re)/10^3, 100, 'k*')
scatter(Re*x3(end, 5)/10^3, (x3(end,2) - Re)/10^3, 100, 'ks')
scatter(Re*dat(end, 5)/10^3, (dat(end,2) - Re)/10^3, 100, 'k^')
xlabel('Range, km', 'fontsize', 16)
ylabel('Altitude, km', 'fontsize', 16)
legend({'Flight Trajectory','$1\textsuperscript{st}$ Stage MECO', '$2\textsuperscript{nd}$ Stage MECO', ['$7.3\ \frac{km}{s}$ Trajectory', newline, ' and 200km altitude']},'location', 'southeast', 'fontsize', 13)

subplot(1,3,3)
hold on
plot(time_accel, accel, 'k', 'linewidth', 2)
scatter(time_accel(1040), max(accel), 100, 'k*')
scatter(time_accel(2900), accel(2900), 100, 'ks')
scatter(time_accel(end), accel(end), 100, 'k^')
xlabel('Time, s', 'fontsize', 16)
ylabel('Acceleration, g''s', 'fontsize', 16)
legend({'Flight Trajectory','$1\textsuperscript{st}$ Stage MECO', '$2\textsuperscript{nd}$ Stage MECO', ['$7.3\ \frac{km}{s}$ Trajectory', newline, ' and 200km altitude']},'location', 'best', 'fontsize', 13)
set(gcf, 'Color', 'w', 'Position', [200 200 1200 500]);
% export_fig('launch_profile.eps')
%   ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

Launch Differential Equation

Listing 5 Matlab Code to numerically approximate launch profiles.

```
function [derivatives] = launchODE(time,parameters,mdot,Thrust_SL,Thrust_vac...
    ,Gimbal,verticalLaunch,fixedPitch,targetPitch)
% GRAVITYTURNODE ODE for a rocket launch subject to a gravity turn
%   Detailed explanation goes here
%   paramters : [Mass, radius, speed, flight-path-angle, true anomaly]
%   mdot : propellant flow rate (assumed positive)
%   Thrust_SL : TOTAL rocket thrust at Sea level conditions
%   Thrust_vac : TOTAL rocket thrust at vacuum conditions
%   Gimbal : Engine gimbal angle (always make 0 for HW1)
%   verticalLaunch : enable vertical launch mode for equations of motions
%   fixedPitch : enable fixed pitch maneuver mode
%   targetPitch : pitch angle for fixed pitch mode [rad]
%If fixedPitch and verticalLaunch are false the code assumes gravity turn

if verticalLaunch && fixedPitch
    error('Cannot use both fix pitch mode and vertical launch mode')
end

% Constants
Re = 6371e3; %Radius of Earth
g0=9.81; %Gravity

%%% Give physical names to input parameters
m = parameters(1); %Mass
r = parameters(2); %Radial position
v = parameters(3); %Speed
psi = parameters(4);%Flight-path-angle
nu = parameters(5); %True anomaly


% Assumptions for vertical launch


%%% Interpolate atmospheric pressure
h_ref = 1e3*[0:9 10:5:25 30:10:80];
P_ref = 1e4*[10.13 8.988 7.95 7.012 6.166 4.405 4.722 4.111 3.565 3.08 2.65...
    1.211 0.5529 0.2549 0.1197 0.0287 0.007978 0.002196 0.00052 0.00011];
Patm = interp1(h_ref,P_ref,parameters(2)-Re,'linear',0);

% Calculate thrust at new altitude
Thrust=Thrust_vac - Patm/P_ref(1)*(Thrust_vac-Thrust_SL);
% dlmwrite('thrust_vals/dat', Thrust/m, '-append');
% dlmwrite('thrust_vals/time', time, '-append');


% Assumptions for this problem
Drag=0;
Lift=0;
AOA=0;

dmdt = -mdot; %Mass
% Radius
if verticalLaunch
    drdt = v;
else
    drdt = v*sin(psi);
end
% Speed
if fixedPitch
    dvdt = -g0*sin(psi) + Thrust/m*cos(targetPitch-psi) - Drag/m;
elseif verticalLaunch
    dvdt = -g0 + Thrust/m*cos(AOA+Gimbal) - Drag/m;
else
    dvdt = -g0*sin(psi) + Thrust/m*cos(AOA+Gimbal) - Drag/m;
end
%Flight path angle
if fixedPitch
    dpsidt = -(g0/v - v/r)*cos(psi)+Thrust/m*sin(targetPitch-psi)/v+Lift/m/v;
```

```
elseif verticalLaunch
    dpsidt = 0;
else
    dpsidt = -(g0/v - v/r)*cos(psi)+Thrust/m*sin(AOA+Gimbal)/m+Lift/m/v;
end
%True anomaly
if verticalLaunch
    dnudt = 0;
else
    dnudt = v*cos(psi)/r;
end

% Derivative vector
derivatives = [dmdt; drdt; dvdt; dpsidt; dnudt];

end
```