High **V**elocity **I**nterstellar and **D**eep Space Pl**A**net **R**econnaissance (VIDAR)

Hunting for Planet 9

Cornell University
MAE 5065

Grace Falanga, William Hintlian, Benjamin Inbar, Alberto Matute

December 7, 2021

Mission Concept Statement

We propose to create a spacecraft that will travel to the theorized location of the undiscovered Planet 9 over a period of approximately 20 years in an attempt to make the first sighting of the planet. During the fly-by, we plan to collect high-resolution imaging as well as data on the composition of Planet 9's atmosphere.

Contents

1 Top-Level Requirements

1.1 Mission Duration

The essential requirements of our mission to discover Planet 9 begin with our mission duration. In making this determination, we found it useful to use New Horizons as a reference, which traveled 40 AU to reach its destination over a time period of about 15 years [1]. While New Horizons provides a good baseline for our mission, various aspects of it need to be changed as we are aiming to explore an object that is at a distance of 380-520 AU, a far larger distance than any manufactured object has ever travelled [2]. Furthermore, a mission of such scale with our current technology could take a large amount of time. If we were to send our spacecraft to Planet 9 at the same speed as New Horizons, it would take over 110 years for the spacecraft to reach the orbit of Planet 9. Even if we were to approximate the speed achieved by Juno, the fastest speed ever achieved by a manufactured object of over 160,000 mph, it would still take over 30 years for the spacecraft to complete its mission. In order to reach Planet 9 on a reasonable timescale, this mission will require very high Δv's making a continuous low thrust system, as described in Section 2.4.3, attractive. By implementing such a system, we will reduce the mission duration to a period of less than 20 years.

1.2 Crew & Spacecraft Requirements

As the proposed mission quite far away and is only looking to perform a fly-by, the spacecraft will be uncrewed. We will only require a single spacecraft as our spacecraft will be built on the ground and placed onto its starting orbit in the initial launch, containing all of the required propulsion systems and scientific instruments for the complete mission.

1.3 Data to Be Collected

Throughout the course of the mission, our spacecraft will collect data that will confirm the presence of Planet 9 and provide more insight on its atmospheric composition during the fly-by. We started looking at New Horizons' instruments as a reference point since it was also interested in investigating Kuiper Belt objects. In our case, the atmospheric investigations will be a bit more limited, allowing us to exclude some instrumentation and decrease the payload mass. Including an imaging system like Ralph, the New Horizons visible/near infrared imager/spectrometer with an increased resolution of 10 μRads/Pixel will be extremely useful. This instrument is capable of mapping surface geology and composition, but can also be used to collect data on atmosphere and surface temperature. The instrument itself has a design ideal for long-duration flyby missions, as it has multiple redundancies and no moving parts other than a single door opened [3]. Additionally, an instrument such as the Long-Range Reconnaissance Imager (LORRI) with a higher 2 μRads/Pixel resolution will be extremely important to collecting images for this mission and assisting with navigation. For New Horizons, this instrument was aiming to collect high-resolution images of the surface morphology and atmosphere of Pluto as well as its giant satellite, Charon [4]. The spacecraft will also carry an instrument for taking ultraviolet wavelength occultation measurements at a resolution of 10 Å FWHM. Since Planet 9 is so far away, we will want to have the highest possible resolution images so we can make the most of this long-distance trip and increase our chances of success.

For atmospheric composition investigations, an instrument like Alice, the UV imaging spectrometer, will be useful in measuring the upper atmospheric composition and temperature of Planet 9. Alice is capable of detecting carbon monoxide, atomic hydrogen, argon, neon, nitrogen, methane, and other hydrocarbons and nitriles. Additionally, we intend to use a radio science package similar to REX (Radio EXperiment) which was utilized on New Horizions. REX is designed to collect temperature and pressure profiles of Pluto's atmosphere to the surface using phase delays in received radio signals from NASA's Deep Space Network, as well as investigating Pluto's ionospheric density [5]. Instruments like these will help our mission to gather useful optical and atmospheric data on Planet 9, while excluding some of the more Pluto-specific probes from the New Horizons payload will allow us to decrease the payload mass.

2 Functional Requirements

2.1 Mission Timeline & Orbital Information

Our mission will consist of 3 mission legs. We will start, departing from Earth, and complete a Hohmann transfer to reach Jupiter's sphere of influence. For our second leg, instead of decelerating to match Jupiter's orbits, we will use this velocity for a gravity assist to gain Δv in order to achieve a faster velocity towards Planet 9. The last leg will consist of an orbit transfer from Jupiter to Planet 9, where we will carry out the goals of the primary mission as outlined in Section 1.3. The Δv required for each leg of the mission is depicted in Table 1, and was calculated in Sections 2.1.1-2.1.3.

Table 1: Complete breakdown of spacecraft Δv

Leg of trip	**Δv (km/s)**
Earth to Jupiter	39.2071
Min. Jupiter to Planet 9	52.5832
Min. Δv required for trip	**91.7745**
Δv by Gravity Assist	52.5832
Δv by Launch Vehicle	13.2
Δv by Spacecraft	26.0071
Total Δv by Spacecraft & Launch Vehicle	**39.2071**

A visualization of the major milestones for our mission is depicted below in Fig. 1, with arrival and departure dates based on our calculations for the minimum Δv in Sections 2.1.1-2.1.3. At the speed we are entering Jupiter's sphere of influence, the gravity assist will take less than one day.

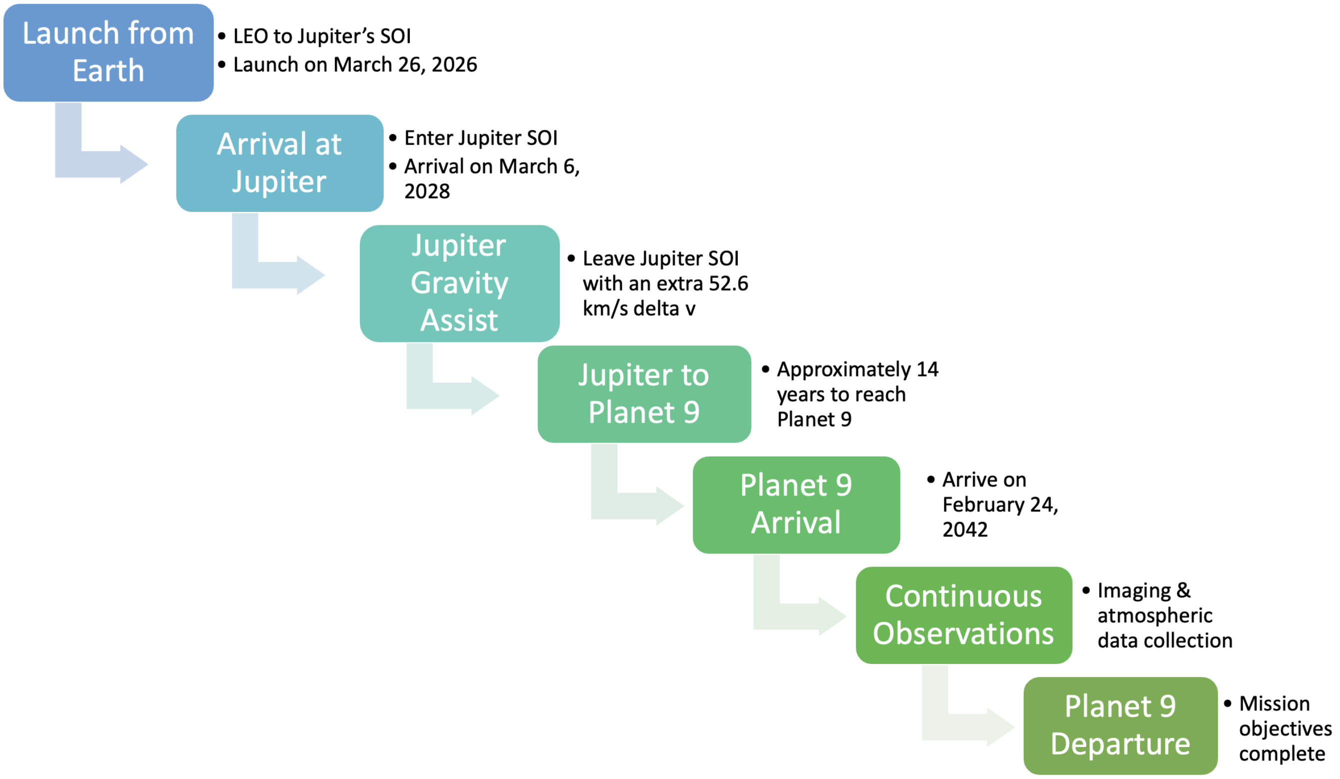

Figure 1: Depiction of major mission events and when they are expected to occur.

2.1.1 Earth to Jupiter

In order to calculate the orbital elements, we utilized an efficient Lambert solver found online [6] for reasons stated in Section 2.2. We assumed a departure date between December 12th 2021 and June 30th 2026, and a minimum transfer time of 1,227 Julian Days. These launch date ranges were decided to allow multiple Earth-Jupiter configurations as well as to show the minimum of the pork chop plot, depicted in Fig. 2. As the position of Planet 9 doesn't vary much, we only worried for the configuration of these two first planets. The transfer time was decided through estimates of other missions, and then proven through our code to give achievable resulting velocities. The Earth-Venus position script was adapted to find the positions and velocities of Earth and Jupiter at our departure and arrival dates, respectively. During this transfer orbit, the central body is the Sun. Performing these calculations resulted in the following:

$$\text{Semi-major axis, } a_t = 3.327123e8 \text{ km}$$

$$\text{Semi-parameter, } l_t = 251667424.444351$$

$$\text{Eccentricity, } e_t = 1.447336$$

$$\text{Transfer Velocity, } v_t = 39.192897 \text{ km/s}$$

2.1.2 Jupiter Gravity Assist

For the gravity assist around Jupiter, we utilized the gravity assist code made for class. In this leg of the mission, the central body is Jupiter. Our arrival date to Jupiter was selected to maximize the velocity after this transfer while still keeping the closest approach larger then the radius of Jupiter. Thus, we were able to estimate our final velocity and the other gravity assist parameters:

$$\text{Turning Angle, } \phi = 1.141555 \text{ rad}$$

$$\text{Closest Approach, } R_p = 70170.860980 \text{ km}$$

$$\text{Eccentricity, } e = 1.850871$$

2.1.3 Jupiter to Planet 9

We once again used the Lambert solver discussed in Section 2.2 for the final leg of the trip. We assumed a transfer time of 6,770 Julian Days, as this allowed gave us an achievable Transfer velocity. We again adapted the Earth-Venus position script to find the positions and velocities of Jupiter and Planet 9 at our departure and arrival dates, respectively. Due to lack of specification as to when the time of periapsis was for Planet 9, we chose one that would orient the planet in the direction of our final velocity for the second leg of the trip. This further minimizes required Δv. During this transfer orbit, the central body has shifted back to the Sun, and performing these calculations resulted in the following:

$$\text{Semi-major axis, } a_t = 3.050557e10 \text{ km}$$

$$\text{Semi-parameter, } l_t = 23742306.141$$

$$\text{Eccentricity, } e_t = 1.000389$$

$$\text{Transfer Velocity*, } v_t = 91.776134 \text{ km/s}$$

2.2 Total Mission Δv Calculations

In order to calculate the total Δv necessary for the journey to Planet 9, we broke the orbit into three major legs. First was the transit to Jupiter, for which we calculated the Δv for a number of departure and arrival dates. These Δvs which can be seen in Fig. 2 were originally calculated using Porkchop_Plotter_E2J.m and the Lambert solver made in class. However, our code was running too slowly as it wasn't optimized for speed, causing the constant iterations of the pork chop plot to run slowly. Thus, we replaced our Lambert solver with one developed by Rody Oldenhuis. This reduced the run time of our code drastically. The Porkchop plots were generated using Matlab scripts provided in Appendix A.1, Appendix A.3 and Appendix A.2.

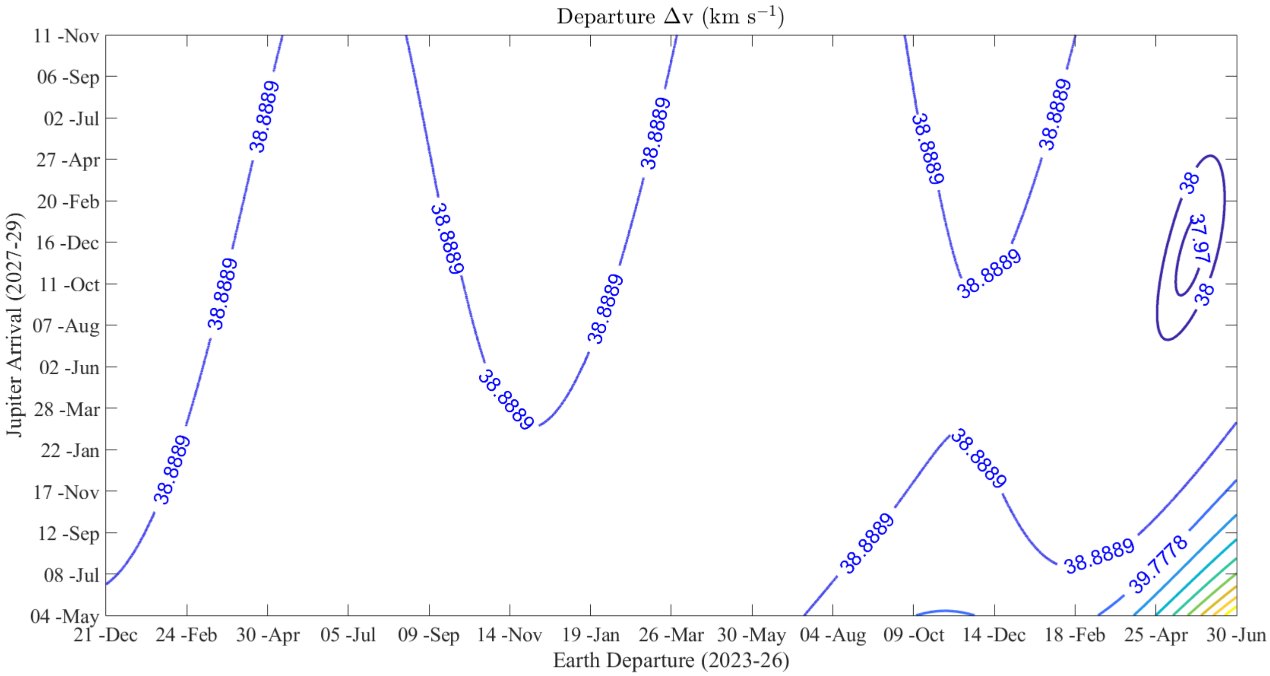

Figure 2: Porkchop plot showing Earth to Jupiter Transfer Δv's

The next leg was the gravitational assist from Jupiter. We developed a script in order to calculate the effects of the gravitational assist shown in Appendix A.3. Our code was optimized around this leg, as the faster the velocity we got from this assist, the more we'd reduced the time from Jupiter to Planet 9. The inputs of this code are $v_{\infty,in}$ and the gravitational parameter. This gravitational parameter is that of the spacecraft and Jupiter ($\mu = 2.824434027777600e - 07AU^3DU^{-2}$), and with the assumption that our initial estimations that the transfer velocity is approximately equivalent to the incoming velocity $v_{\infty,in}$. This incoming velocity was the result of our Earth to Jupiter Lambert solver calculations, a transfer velocity of $v_t = 39.192897$ km/s. Due to this assist, we are able to get the Δv required to get to Planet 9. Thus, our resulting parameters indicate a total mission Δv of approximately 91.7745 km/s, with only 39.2071 km/s needed to get to Jupiter. This components of this breakdown are shown in Table 1.

Our last leg was from Jupiter to Planet 9. The Δv's for this transfer can be seen in Fig. 3. Interestingly, these contours are fairly vertical. This is most likely due to the relatively small change in position of Planet 9 over the course of the arrival window. With an orbital period on the order of 7,000 years, its position is relatively constant over the timescales of the mission.

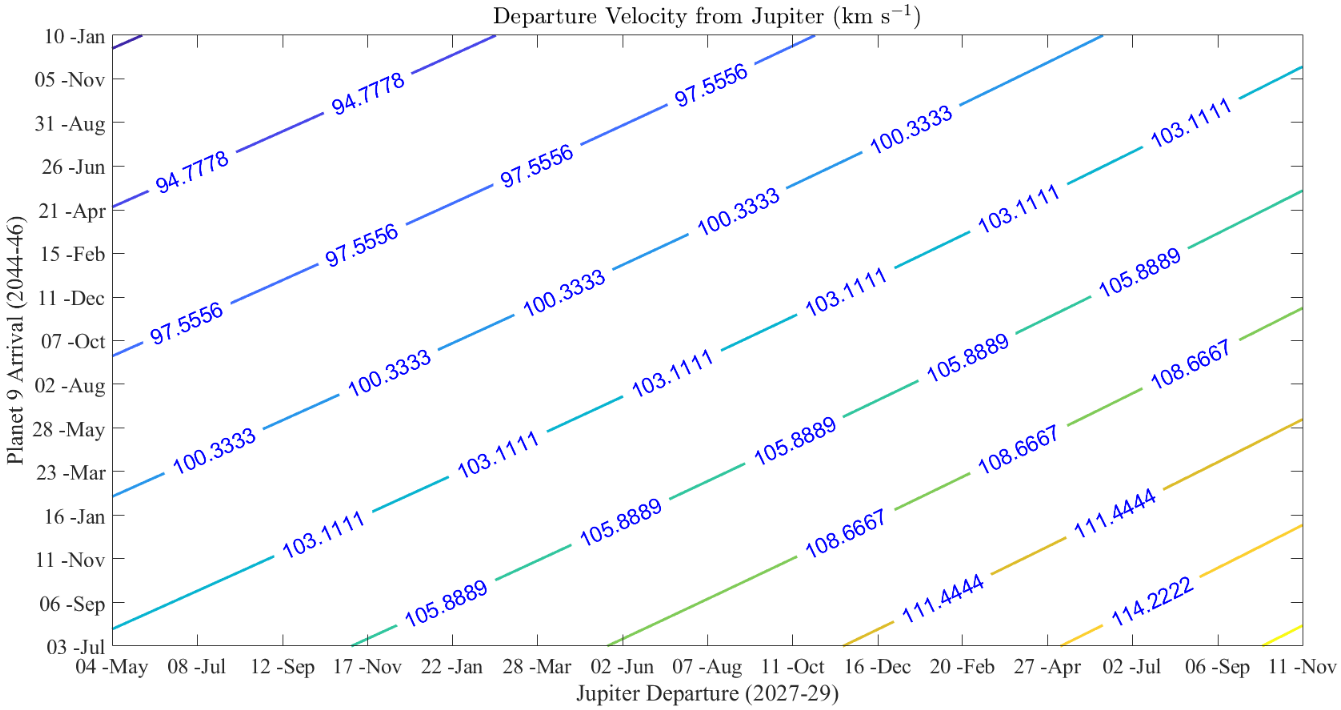

Figure 3: Porkchop plot showing Jupiter to Planet 9 Transfer velocity

Taking the minimum transfer Δv for each of these transfers that provides enough velocity to reach Planet 9, we can find that the total mission Δv is around 40 km/s. The departure date from Earth will be March 26th 2026, arriving at Jupiter's influence at March 6th 2028 and after the assist, finally reaching Planet 9 on February 24th 2042.

These Δv's were calculated assuming that the spacecraft at time of departure from Earth and Jupiter was traveling with the velocity of the body. Some details in the calculations still remain, so the values for our Δv could vary slightly, but the value above is a fairly accurate estimate.

2.3 Effect of Perturbations

Perturbation forces are mostly negligible for our mission as currently planned. We neglect perturbations due to drag as we plan to launch from outside of Earth's atmosphere. We also neglect all spherical harmonic perturbations from other celestial bodies as our mission only orbits the sun on its way to the Planet 9 flyby. The short time spent in Jupiter's sphere of influence during the gravity assist is not enough to noticeably affect the mission's orbital elements beyond the gravity assist.

Third body forces may affect all of our orbital elements, but we are not yet able to compute the magnitude of their effect as we do not know the exact timing and trajectory of our mission. In future analysis, the mission may consider third body forces from Mars, Uranus, Neptune, Pluto, or other large celestial masses.

Solar radiation pressure is expected to cause a perturbing force in the radial direction of orbit. Although our spacecraft will have a relatively small surface area and large mass relative to that surface area (compared to a solar sail), the length of the mission led the team to analyze the potential effects of solar radiation pressure on our orbital elements. We found that over the course of our flight from Earth to Planet 9, solar radiation pressure will cause eccentricity to increase on the order of 10^{-8}. As a result of this change, Semi-major Axis, Inclination, Argument of Periapsis, and Longitude of the Ascending Node will also change by similar orders of magnitude. The script used for this estimation is included in Appendix A.7.

2.4 Launch Vehicle Selection

2.4.1 Payload Mass Determinations

To determine the proper launch vehicle for our mission, we need to establish how massive our payload will be. In determining this amount, we started by gathering an understanding of what the minimum requirements are given our scientific objectives. This baseline started with the New Horizons payload, as we will be utilizing similar data collection and transmission equipment. In the New Horizons mission, due to the capabilities of the propulsion system and launch rockets, the entire spacecraft mass was required to be less than 480 kg, with less than 50 kg for scientific instruments. In the end, the actual scientific payload was under 30 kg. Since we are expecting to exclude three of the scientific instruments more focused on Pluto-specific investigations, our scientific payload will be below 25 kg, as detailed in Table 2 [5]. This estimate served as a starting point for determining what sorts of propulsion would be feasible given how far we need to go. From there, we looked to the Rocket Equation (Eqn. 1) in order to calculate the available dry mass of our spacecraft,

$$\Delta v = v_{\text{eff}} \ln \frac{m_0}{m_f} \tag{1}$$

where v_{eff} is the the effective exhaust velocity of the rocket. Specific impulse is related to the exhaust velocity v_{eff} through the equation, $I_{sp} \equiv \frac{v_{\text{eff}}}{g_0}$, where g_0 is the standard gravitational acceleration at Earth's surface.

Table 2: Breakdown of the mass requirements for each scientific instrument.

Instrument	Mass (kg)
Alice UV Spectrograph	4.15
Ralph visible & IR Imager	10.67
REX	0.1
LORRI	8.59
Total Mass	**23.51**

We started by assuming a C_3 of 300s from the SLS as a launch rocket could be achieved with a payload mass of 800 kg [7], and that the upper limit of our required Δv would be 40 km/s total (Table 1). By using the definition of the characteristic energy, C_3:

$$C_3 = v^2 - v_{\text{esc}}^2 \tag{2}$$

with v_{esc} indicating the Earth's escape velocity (11.2 km/s) [8], and v indicating the velocity of the spacecraft. We used Eqn. 2 to calculate the velocity provided by a launch vehicle putting the spacecraft on an orbit with a C_3 of 300 km^2/s^2. Using the Earth to Jupiter Δv requirement of approximately 40 km/s shown in Table 1, we calculated that the launch vehicle could provide us with an initial Δv of 13.2 km/s. Our spacecraft would need to provide an additional 26.8 km/s with its on-board propulsion system. This Δv value was used in all of the Rocket Equation (Eqn. 1) calculations.

Table 3: Complete breakdown of spacecraft mass components, including total wet and dry masses.

Payload Component	Mass (kg)
Scientific Instruments	23.51
Structural Components	25
RTG	225
RCS System	6.31
NSTAR Ion Engine	50
Total Spacecraft Dry Mass	**324.82**
Hydrazine for RCS	35
Xenon for NSTAR	470.18
Total Spacecraft Wet Mass	**800**

By using electric propulsion rather than a traditional chemical propulsion system, the spacecraft is capable of providing 26.8 km/s of Δv. Section 2.4.3 outlines the calculations required to find the xenon fuel mass necessary to achieve this velocity. The Reaction Control System (RCS) is composed of 12 monopropellant thrusters each with a mass of .38 kg, hydrazine and a hydrazine tank whose mass is assumed to be 5% of the fuel mass [9]. The RCS Fuel mass was selected by adding a 16% margin to New Horizon's RCS fuel supply [10]. Electric propulsion was selected because of the high required Δvs; if we were to use LH/LOX combination fuel, which is known as a chemical propellant with one of the highest specific impulses possible ($I_{\text{sp}} = 451s$), would result in space for a dry mass of only 1.87 kg given our Δv requirements. By moving away from chemcial propulsion, we are able to propel a rocket with a dry mass of 325 kg, as detailed in Table 3. This provides the space for all of the required equipment to complete our mission objects.

2.4.2 Potential Launch Vehicles

In order to reach Planet 9, the spacecraft will have to leave Earth with a C_3 significantly greater than any other mission ever launched. This narrows down the available launch vehicles to Super-Heavy lift rockets of which there are only two viable options: NASA's Space Launch System (SLS), and SpaceX's Starship/Superheavy system. Each of these two systems would result in a different departure trajectory from Earth to Planet 9. Ultimately the SLS was selected over the SpaceX vehicle due to lower risk and complexity.

Utilizing the SLS, the spacecraft would be placed on a direct injection to Planet 9. In order to reach a C_3 in the 300 - 450 $\text{km}^2\ \text{s}^{-2}$ the mission would require the use of an SLS Block 2 with a Centaur third stage, and a Star 48BV as the fourth stage [7]. While the core stage of the SLS is yet untested, it makes use of proven flight hardware. The 4 main engines are RS-25s which flew hundreds of missions with the Space Shuttle, the SRBs are also carried over from the Shuttle program meaning that while the fully integrated core stage has never flown, the risk is certainly lower due to the well understood components [11]. While there is not yet a price tag on the SLS Block 2 rocket, the current estimation for SLS Block 1 is $2 billion per launch, significantly more than any other launch vehicle that currently exists [12].

Starship and Superheavy would potentially rely on a very different trajectory than SLS. Rather than a direct injection to Planet 9, Starship would launch from Earth into a refueling orbit and then execute an injection to Planet 9 [13]. According to the Starship user guide, the system will not offer direct injection for lunar and martian missions, however as this system is still in its prototype phase, direct injections may become an option [13]. The payload will be launched by the system into low earth orbit, where the upper Starship stage will be refueled. This in-orbit refueling enables sending a very heavy spacecraft somewhere relatively close (like Mars) or a very light spacecraft very far and very fast [13]. While no price has been set, SpaceX founder Elon Musk has claimed a cost of $2 million per launch in a fully reusable configuration [14]. Given that the upper stage would be placing the spacecraft on a deep space trajectory it is unlikely that it would be recovered. The expended first stage will most likely add cost to the total launch (and the $2 million figure is likely to change), but will likely remain significantly below an SLS Block 2 launch. There is also the possibility of using Starship for a direct injection to Planet 9, however SpaceX has not published information or specifications for this type of launch.

Starship is an entirely new vehicle with hardware that has a significantly shorter service history than the SLS. The first test flights of the upper stage which was also the first test flight of the methane powered Raptor engines, occurred in late 2020 with the first successful flight occurring in March of 2021 [14]. The Superheavy first stage has yet to complete its first test flight, and will not fly until 2022 [15]. While this indicates a lower technological readiness level at present, it also demonstrates that SpaceX is moving at a far faster pace than NASA. The Starship Superheavy system may be available for launch sooner than SLS if the first full stack test is successful.

Both launch systems have the potential to be viable launch systems for the mission to Planet 9. NASA's SLS while potentially less risky than Starship is three orders of magnitude more expensive. Starship on the other hand has serious risks associated with it using both less proven hardware, and a very risky in orbit refueling maneuver (something which has never been attempted before). There is also less available data on Starship, making it more difficult to determine the kinds of orbits and transfers that it can provide. One potential benefit to Starship is increased payload mass with SpaceX claiming that Starship could deliver 100+ metric tons to Mars [13]. This may allow a larger payload to be sent to Planet 9 then SLS would be capable of.

2.4.3 Propulsion Systems

To achieve the required velocity before beginning the gravity assist maneuver with Jupiter, the spacecraft's onboard propulsion system must provide an additional 26 km/s of Δv shown in Table 1. The mission will use an ion thruster similar to that used on the Deep Space 1 mission. Developed by the NSTAR program, the combined mass of the thruster and its required support systems is approximately 50 kg [16].

Due to low solar insolation in the outer solar system, solar panels are not a viable option to power the ion engine. A radically more powerful Radioisotope Thermoelectric Generator (RTG) will be developed for this mission. The power system must provide 2.3 kW of power during the trip to Jupiter to operate the thruster at its maximum I_{sp} of 3,120 s. At approximately 57 kg, the RTG for New Horizons provided 250 W of power at launch with an annual decay of 1.25% [17]. The Planet 9 mission will require a more powerful version of the RTG containing a larger supply of plutonium to meet the thruster's requirements. Currently there have been no RTGs made to produce enough power for our mission. We require a modern RTG or RTG array which can produce a total of 2.3 kW without exceeding 225 kg. The system will first power the NSTAR thruster on the way to Jupiter. Then, it will power scientific instruments, communication systems, and attitude control systems near Planet 9. To generate 2.3 kW of solar power as far away from the sun as Jupiter, the spacecraft requires 150 m^2 of solar panels operating at 30% efficiency. The script used to calculate solar panel sizes is included in Appendix A.8 Assuming the mass of the solar panels is approximately 2 kg/m^2 [18], the total solar panel mass sums to at least 300 kg, as structural and other supporting mass is not included in the estimate. This makes solar panels not suitable for our mission. Due to the low-thrust nature of the thruster and its relatively high input power requirement, a battery to store enough energy to power the system would be far too massive to use.

In total, the propulsion system mass sums to less than 225 kg. Given a payload of approximately 30 kg and assuming approximately 25 kg for structural and other mass, the spacecraft's total dry mass is approximately 330 kg. The SLS launch vehicle supports a total payload of approximately 800 kg for our mission. Therefore, the available fuel mass is 470 kg of xenon (See Table 3). Assuming the ion thruster provides its maximum I_{sp} of 3,120 s with 2.3 kW power input [16], we apply Eqn. 1 to compute the available Δv. The propulsion system offers a total Δv of at least 27.1 km/s if all of the xenon fuel is consumed, exceeding the necessary Δv and ensuring there will be fuel to perform course correcting burns or maneuver to a secondary mission.

3 Attitude Control Performance Requirements & Constraints

3.1 Restrictions on Pointing

In order to achieve mission goals, it is critical that the spacecraft be able to receive instructions from and transmit data to Earth. This will be by far the most critical pointing requirement for the mission to Planet 9, and thus the primary constraint on the Attitude Determination and Control System (ADCS). In order for New Horizons to maintain a 42 db connection with Earth at Pluto, it was required to maintain a pointing accuracy of ±0.3° [19]. An amplitude of 42 db at earth was necessary to maintain 600 bps of downlink capability, and corresponds to a signal cone with a diameter of approximately 0.21 AU at Earth [19].

For a mission to Planet 9 with an orbital radius of almost 10 times that of Pluto, significantly stricter pointing requirements must be enforced in order to achieve similar downlink speeds. Alternatively, a much more powerful antenna could be used to transmit data back to Earth while maintaining the same pointing requirements as New Horizons. However this option has been ruled out due to the large amount of power this would require. This would create a signal cone with a diameter of 1.9 AU which would be a significant waste of power. This is simply not a feasible solution thus highly precise pointing must be implemented. This mission will assume a communication system capable of producing signal cone with a diameter at Earth equal to that of New Horizons (0.21 AU) with a similar 42 db to maintain the 600 bps downlink speed. This corresponds to a cone with a half angle of 275.581 µRad (0.016°) as shown in Fig. 4, using the same signal cone diameter of New Horizons.

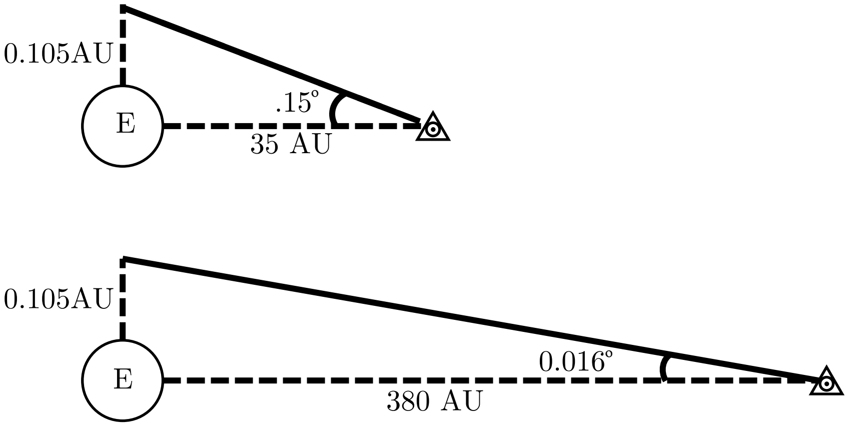

Figure 4: Signal cones of New Horizons as compared to the proposed signal cone of the Planet 9 mission.

In order to meet these pointing requirements, this mission will require very high precision attitude determination as well as attitude control systems in order to achieve this pointing. For attitude determination, we will implement high accuracy star trackers in order to determine our position. Much like the New Horizons spacecraft we will spin stabilize the spacecraft during cruise portions of flight with the axis of spin through the center of the high gain antenna and pointed at Earth. Because of this we will make use of the same APL Autonomous Star Tracker (ASTR) system, which had been modified for New Horizons to include a spin mode allowing it to determine spacecraft position while in the spin stabilized configuration [20]. We will also uses MEMs IMUs to propagate the spacecraft attitude between star tracker refreshes to have a more continuous description of the spacecrafts attitude. This will enable the spacecraft to determine its attitude with enough accuracy to maintain the very small pointing envelope necessary for communication. The star trackers and IMUs on New Horizons allowed it to determine its position to within ±471 µRad [10]. Our mission with a pointing requirement of ±558.5 µRad does not necessarily need a system more accurate than this, however, with over 15 years having passed since the launch of New Horizons advancements in technology (especially in IMUs) will allow even higher precision. The spacecraft will then use an RCS to provide full 3-axis control in order to accurately point where needed. The RCS system will consist of 12 1 N MONARC-1 monopropellant thrusters manufactured by MOOG. These will mounted on opposite sides of the spacecraft in firing pairs in order to provide attitude control.

This ADCS system will provide the necessary precision to communicate with Earth from 100s of AU away. It will also provide more than adequate control of the spacecraft for its encounter with Planet 9, where it will have to point cameras and scientific instruments to collect data.

3.2 Restrictions on Maneuvering Accelerations

As this will be an uncrewed mission the restrictions on maneuvering accelerations will be significantly more relaxed than they would be on a crewed mission. The main concern is with regards to acceleration is the initial launch and tranjulian injection. This is the only time where the spacecraft will be undergoing high acceleration and where the acceleration cannot be reasonably reduced. Thus the hardware on the spacecraft must be designed to be able to withstand the loading from launch.

Over the remainder of the mission, the spacecraft will be propelled by ion thrusters that output 19 to 92 mN of thrust resulting in a possible acceleration range of 0.02375 to 0.115 $\frac{mm}{s^2}$. The reaction control system will likewise not be providing significant acceleration to the spacecraft and are likely not to be a concern. All equipment present on the spacecraft will have been built and capable of operation in Earth gravity. The RCS and Ion thrusters will not be applying accelerations on a magnitude near 1 g and so there is little cause for acceleration induced damage.

4 Risk Identification & Mitigation

In any mission, there are a host of risks involved, but these risks often become more complex to mitigate for deep space missions. As a result, it is crucial that we evaluate all potential risks and set up a plan should these events occur. Additionally, it is important to evaluate the severity of each risk present based on its likelihood and impact. We have done this for each risk we identified based on the standard Risk Matrix shown in Table 4. The severity of each risk will serve to prioritize their mitigation should it become necessary.

Table 4: Risk Matrix used to determine the severity of each potential risk

		Impact		
		Low	Medium	High
Likelihood	High	Medium	High	Very High
	Medium	Low	Medium	High
	Low	Low	Low	Medium

The first risk we have identified is the possibility that the location of Planet 9 or the details of its orbit are wrong, so we miss the planet entirely. While this occurrence would have a very high impact and prevent the completion of our mission objectives, much research has gone into pinning down Planet 9's location, so we do not anticipate it having a high likelihood. In the event that it did occur, we would adjust the mission goals to potentially explore and collect data on other Kuiper Belt objects and the general region.

Another potential risk is that a launch event causes damage to the RTG. While the impact to the mission would be high, the RTG is a very well-developed technology, so we expect the likelihood of such an instance to be very low. In order to mitigate any potential risks, a safety analysis will be performed by the Department of Energy, and will receive a nuclear safety launch approval as is standard with any launch containing an RTG [21].

In terms of the launch window, if weather causes us to miss our launch window, this would definitely cause a medium-to-high impact. While the fuel we have on board will provide us with some excess Δv than what is strictly necessary, if we miss our launch window, the required Δv may increase significantly, or it could greatly draw out the length of the mission, due to our orbital maneuvers. The likelihood of this occurring depends partly on where we are launching from, but is an issue that is dealt with routinely, so we do not anticipate it catastrophically affecting our mission.

One particularly critical risk concern is loss of contact with the spacecraft. This concern is mitigated by the passive spin stabilization of the spacecraft with its high gain antenna pointed towards Earth. Once out of the thrust phase of flight, the spacecraft will remain in this spin stabilized mode until it is required to perform actions such as imaging Planet 9 or performing trajectory correction maneuvers. After these actions the spacecraft will autonomously return to its Earth pointing spin stabilized state. By having the spacecraft programmed to autonomously return to this configuration when not performing actions, it will remain in contact with Earth.

In order to mitigate any potential issues with our instrumentation, the design will include multiple redundancies. Additionally, since all of our scientific instruments are currently active on New Horizons, we can analyze what problems they have encountered so far and create a plan based on what was successful in their mitigation of issues.

In terms of dust interference, we can utilize a similar risk mitigation plan as New Horizons, which relied on pre-programming the trajectory imaging data from the New Horizons mission to predict with better accuracy potential issues that may arise on our trajectory. Although our trajectories are not the same, we will be travelling through similar areas, and in conjunction with their on-craft technique will serve to avoid any severe damage from dust [22]. It is highly unknown whether or not dust will be a significant issue due to the lack of knowledge about the Planet 9 system. We do not know if the spacecraft will encounter moons or dust clouds and so it is critical to plan as though we will.

5 References

[1] NASA, *New horizons: In depth*, `https://solarsystem.nasa.gov/missions/new-horizons/in-depth/`, Accessed: 2021-10-08, Oct. 2020.

[2] M. E. Brown and K. Batygin, "The orbit of planet nine," *The Astronomical Journal*, vol. 162, no. 5, p. 219, 2021.

[3] D. C. Reuter, S. A. Stern, J. Scherrer, *et al.*, "Ralph: A visible/infrared imager for the new horizons pluto/kuiper belt mission," *Space Science Reviews*, vol. 140, no. 1, pp. 129–154, 2008.

[4] A. F. Cheng, H. Weaver, S. Conard, *et al.*, "Long-range reconnaissance imager on new horizons," *New Horizons*, pp. 189–215, 2009.

[5] H. Weaver, W. Gibson, M. Tapley, L. Young, and S. Stern, "Overview of the new horizons science payload," *New Horizons*, pp. 75–91, 2009.

[6] R. Oldenhuis, *Robust solver for lambert's orbital-boundary value proble*, version 1.4, May 2, 2020. [Online]. Available: `https://github.com/rodyo/FEX-Lambert/releases/tag/v1.4`.

[7] R. Stough, K. F. Robinson, J. B. Holt, D. A. Smith, W. D. Hitt, and B. A. Perry, "NASA's space launch system: Capabilities for ultra-high c3 missions," *Bulletin of the AAS*, vol. 53, no. 4, Mar. 18, 2021. DOI: `10.3847/25c2cfeb.170da7f3`. [Online]. Available: `https://baas.aas.org/pub/2021n4i451` (visited on 11/17/2021).

[8] T. Kadono, T. Sakaiya, Y. Hironaka, *et al.*, "Impact experiments with a new technique for acceleration of projectiles to velocities higher than earth's escape velocity of 11.2 km/s," *Journal of Geophysical Research: Planets*, vol. 115, no. E4, 2010.

[9] Moog Inc. "Spacecraft thrusters." (), [Online]. Available: `https://www.moog.com/products/propulsion-controls/spacecraft/thrusters.html` (visited on 12/07/2021).

[10] G. H. Fountain, D. Y. Kusnierkiewicz, C. B. Hersman, *et al.*, "The new horizons spacecraft," *Space Science Reviews*, vol. 140, no. 1, pp. 23–47, Oct. 2008, ISSN: 0038-6308, 1572-9672. DOI: `10.1007/s11214-008-9374-8`. arXiv: `0709.4288`. [Online]. Available: `http://arxiv.org/abs/0709.4288` (visited on 12/07/2021).

[11] L. Mohon. "Space launch system (SLS) overview," NASA. (Mar. 16, 2015), [Online]. Available: `http://www.nasa.gov/exploration/systems/sls/overview.html` (visited on 11/17/2021).

[12] R. T. Vought, *Letter to the chair and vice chair of the senate appropriations committee with respect to 10 of the FY 2020 annual appropriations bills*, Letter. [Online]. Available: `https://www.whitehouse.gov/wp-content/uploads/2019/10/shelby-mega-approps-10-23-19.pdf` (visited on 11/17/2021).

[13] J. Jensen, *Starship users guide revision 1.0*, Mar. 2020. [Online]. Available: `https://www.spacex.com/media/starship_users_guide_v1.pdf` (visited on 11/17/2021).

[14] M. Bender, "SpaceX's starship could rocket-boost research in space," *Scientific American*, Sep. 17, 2021. [Online]. Available: `https://www.scientificamerican.com/article/spacexs-starship-could-rocket-boost-research-in-space/`.

[15] M. Wall. "FAA to wrap up SpaceX starship environmental assessment by dec. 31 | space." (), [Online]. Available: `https://www.space.com/faa-spacex-starship-starbase-review-deadline` (visited on 11/17/2021).

[16] M. Wade. "NSTAR." (), [Online]. Available: `http://www.astronautix.com/n/nstar.html` (visited on 11/17/2021).

[17] "The heat is on: How new horizons got its power | GE news." (), [Online]. Available: `https://www.ge.com/news/reports/the-heat-is-on-how-new-horizons-got-its-power` (visited on 11/17/2021).

[18] Spectrolab Inc. [Online]. Available: `https://www.spectrolab.com/DataSheets/Panel/panels.pdf`.

[19] R. Schulze and S. Hill, "The new horizons high gain antenna: Reflector design for a spin-stabilized bus at cryogenic temperatures," in *2004 IEEE Aerospace Conference Proceedings (IEEE Cat. No.04TH8720)*, ISSN: 1095-323X, vol. 2, Mar. 2004, 966–974 Vol.2. DOI: `10.1109/AERO.2004.1367697`.

[20] G. D. Rogers, M. R. Schwinger, J. T. Kaidy, *et al.*, "Autonomous star tracker performance," *Acta Astronautica*, vol. 65, no. 1, pp. 61–74, Jul. 1, 2009, ISSN: 0094-5765. DOI: `10.1016/j.actaastro.2009.01.045`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S0094576509000757` (visited on 12/03/2021).

[21] *New horizons launch press kit*, Dec. 15, 2005.

[22] D. S. Mehoke, D. A. Seal, T. S. Mehoke, *et al.*, "The new horizons pluto fly-by dust hazard assessment process," in *2017 IEEE Aerospace Conference*, Mar. 2017, pp. 1–12. DOI: `10.1109/AERO.2017.7943630`.

A Appendix

A.1 Porkchop Plotter Script for First Leg

Porkchop_Plotter_E2J.m: Script for building porkchop plots.

```
clc; clear; close all;

mus = 2.9591220828559093e-4; %sun Gm in AU^3/day^2
kmAU = 149597870.700; %1 AU in km
%departures
tdep0 = juliandate(datetime('2023-12-21'));
tdepf = juliandate(datetime('2026-06-30'));
tdep = tdep0:1:tdepf;
%arrivals
tarr0 = tdep0 + 1230;
tarrf = tdepf + 1230;
tarr = tarr0:1:tarrf;

[d,r] = meshgrid(tdep ,tarr);
dts = r-d;
exitflags = zeros(length(tdep),length(tarr));
dv1s = zeros(length(tdep),length(tarr));
MV1 = zeros(length(tdep),length(tarr),3);
MV2 = zeros(length(tdep),length(tarr),3);
MRE = zeros(length(tdep),length(tarr),3);
MRJ = zeros(length(tdep),length(tarr),3);
MVE = zeros(length(tdep),length(tarr),3);
MVJ = zeros(length(tdep),length(tarr),3);
mu = 1.32712440018e11; %heliocentric gravitational constant

for j = 1: length(tdep)
    fprintf (" Iteration %d of %d\n",j,length(tdep))

    for k = 1: length(tarr)
        dtt = tarr(k) - tdep(j);
        [r1 ,r2 ,v1 ,v2] = EarthJupiterPosVel(tdep(j),dtt);
        [V1, V2, ~ ,exitflags(j,k)] = lambert(r1'*kmAU, r2'*kmAU, dtt, 0, mu);

        MV1(j,k,1) = V1(1); MV1(j,k,2) = V1(2); MV1(j,k,3) = V1(3);
        MV2(j,k,1) = V2(1); MV2(j,k,2) = V2(2); MV2(j,k,3) = V2(3);
        MVE(j,k,1) = v1(1); MVE(j,k,2) = v1(2); MVE(j,k,3) = v1(3);
        MVJ(j,k,1) = v2(1); MVJ(j,k,2) = v2(2); MVJ(j,k,3) = v2(3);
        MRE(j,k,1) = r1(1); MRE(j,k,2) = r1(2); MRE(j,k,3) = r1(3);
        MRJ(j,k,1) = r2(1); MRJ(j,k,2) = r2(2); MRJ(j,k,3) = r2(3);

        dv1t = norm(V2 - V1);
        dv1s(j,k) = dv1t;
    end
end

%% Plot the contour
%lets do 3 ticks on the departure axis
xticks = linspace(tdep0 ,tdepf ,15);
xticklabels = cell(size(xticks ));
```

```
for j = 1: length(xticks)
xticklabels{j} = char(datetime(xticks(j),...
'ConvertFrom','JD','Format','dd -MMM'));
end
%and 5 on the y axis
yticks = linspace(tarr0 ,tarrf ,15);
yticklabels = cell(size(yticks ));
for j = 1: length(yticks)
yticklabels{j} = char(datetime(yticks(j),...
'ConvertFrom','JD','Format','dd -MMM'));
end
%define contour levels to plot
lvls = [ceil(min(dv1s(dv1s >0))*100)/100 ,3.75 ,4 ,4.5 ,5 ,6 ,8 ,10 ,15 ,25 , ...
floor(max(dv1s (:)))];
figure (1)
clf
[C1 ,h1] = contour(tdep ,tarr ,dv1s.','Linewidth',2);
clabel(C1,h1 ,'Color','b','FontSize',16)
hold on
hold off
set(gca ,'FontName','Times','FontSize',16,'XTick',xticks ,...
'XTickLabel',xticklabels ,'YTick',yticks ,'YTickLabel',yticklabels)
xlabel('Earth Departure (2023-26)')
ylabel('Jupiter Arrival (2027-29)')
title('Departure $\Delta$v (km s$^{-1}$)','Interpreter','LaTeX')
set(gca ,'FontName','Times','FontSize',16,'XTick',xticks ,...
'XTickLabel',xticklabels ,'YTick',yticks ,'YTickLabel',yticklabels)

save('Earth2Jupiter_ContourData.mat','tdep','tarr','dv1s','MV1','MV2','MRE','MRJ','MVE','MVJ');
```

A.2 Porkchop Plotter Script for Last Leg

Porkchop_Plotter_J2P9.m: Script for building porkchop plots.

```
clc; clear; close all;
%Earth?Venus Porkchop plot for 2021 launch opportunity
mus = 2.9591220828559093e-4; %sun Gm in AU^3/day^2
kmAU = 149597870.700; %1 AU in km
%the departure
tdep0 = juliandate(datetime('2023-12-21')) + 1230;
tdepf = juliandate(datetime('2026-06-30')) + 1230;
tdep = tdep0:1:tdepf;
%the arrival
tarr0 = tdep0 + 7000;
tarrf = tdepf + 7000;
tarr = tarr0:1:tarrf;

load('JupiterAssist_ContourData.mat')

[d,r] = meshgrid(tdep ,tarr);
dts = r-d;
exitflags = zeros(length(tdep),length(tarr));
```

```
dv1s = zeros(length(tdep),length(tarr));

MV1_2 = zeros(length(tdep),length(tarr),3);
MRP9 = zeros(length(tdep),length(tarr),3);
MVP9 = zeros(length(tdep),length(tarr),3);

mu = 1.32712440018e11; %heliocentric gravitational constant
for j = 1: length(tdep)
    fprintf (" Iteration %d of %d\n",j,length(tdep))

    for k = 1: length(tarr)
        dtt = tarr(k) - tdep(j);
        [r1, r2, v1,v2] = JupiterPlanet9PosVel(tdep(j),dtt);
        [V1, V2, ~ ,exitflags(j,k)] = lambert(r1'*kmAU, r2'*kmAU, dtt, 0, mu);

        MV1_2(j,k,1) = V1(1); MV1_2(j,k,2) = V1(2); MV1_2(j,k,3) = V1(3);
        MVP9(j,k,1) = v2(1); MVP9(j,k,2) = v2(2); MVP9(j,k,3) = v2(3);
        MRP9(j,k,1) = r2(1); MRP9(j,k,2) = r2(2); MRP9(j,k,3) = r2(3);

        dv1t = norm(V2 - V1);
        dv1s(j,k) = dv1t;
    end
end
%% Plot the contour
%lets do 3 ticks on the departure axis
xticks = linspace(tdep0 ,tdepf ,15);
xticklabels = cell(size(xticks ));
for j = 1: length(xticks)
xticklabels{j} = char(datetime(xticks(j),...
'ConvertFrom','JD','Format','dd -MMM'));
end
%and 5 on the y axis
yticks = linspace(tarr0 ,tarrf ,15);
yticklabels = cell(size(yticks ));
for j = 1: length(yticks)
yticklabels{j} = char(datetime(yticks(j),...
'ConvertFrom','JD','Format','dd -MMM'));
end
%define contour levels to plot
minlvl = min(min(dv1s));
maxlvl = max(max(dv1s));
lvls = round(linspace(minlvl,maxlvl,10),2);
lvls = [round(minlvl,1),lvls,round(maxlvl,1)];
figure (1)
clf
[C1 ,h1] = contour(tdep,tarr,dv1s.',lvls,'Linewidth',2);
clabel(C1,h1 ,'Color','b','FontSize',16)
set(gca ,'FontName','Times','FontSize',16,'XTick',xticks ,...
'XTickLabel',xticklabels ,'YTick',yticks ,'YTickLabel',yticklabels)
xlabel('Jupiter Departure (2027-29)')
ylabel('Planet 9 Arrival (2044-46)')
title('Departure $\Delta$v (km s$^{-1}$)','Interpreter','LaTeX')
set(gca ,'FontName','Times','FontSize',16,'XTick',xticks ,...
'XTickLabel',xticklabels ,'YTick',yticks ,'YTickLabel',yticklabels)

save('Jupiter2Planet9_ContourData.mat','dv1s','MV1_2','MRP9','MVP9');
```

A.3 Porkchop Plotter Script for Gravity Assist

Porkchop_Plotter_JAssist.m: Script for building porkchop plots.

```
clc; clear; close all;

mus = 2.9591220828559093e-4; %sun Gm in AU^3/day^2
kmAU = 149597870.700; %1 AU in km

load('Earth2Jupiter_ContourData.mat')

deltavs = zeros(length(tdep),length(tarr));
finalV = zeros(length(tdep),length(tarr),3);

mu = 1.32712440018e11; %heliocentric gravitational constant
mu_j = 1.2668e8;
for j = 1: length(tdep)
    fprintf (" Iteration %d of %d\n",j,length(tdep))

    for k = 1: length(tarr)
        V2 = zeros(3,1);
        V1(1) = MV1(j,k,1); V1(2) = MV1(j,k,2); V1(3) = MV1(j,k,3);
        V2(1) = MV2(j,k,1); V2(2) = MV2(j,k,2); V2(3) = MV2(j,k,3);
        r2(1) = MRJ(j,k,1); r2(2) = MRJ(j,k,2); r2(3) = MRJ(j,k,3);
        [phi, ~, ~, deltavs(j,k)] = gravityassist(mu_j,V1);
        Vnew = Rz(phi)*V1'/norm(V1);
        Vnew = Vnew*(deltavs(j,k) + norm(V1));
        finalV(j,k,1) = Vnew(1); finalV(j,k,2) = Vnew(2); finalV(j,k,3) = Vnew(3);
    end
end

% [M,I1] = max(finalV);
% [MaxV,I2] = max(M);
% j = I1(I2);
% k = I2;
% MaxV1(1) = MV1(j,k,1); MaxV1(2) = MV1(j,k,2); MaxV1(3) = MV1(j,k,3);
% Maxtarr = tarr(k);
% Maxtdep = tdep(j);
% Maxtarr_char = char(datetime(Maxtarr,'ConvertFrom','JD','Format','dd/MMM/yyyy'));
% Maxtdep_char = char(datetime(Maxtdep,'ConvertFrom','JD','Format','dd/MMM/yyyy'));
% [r1 ,r2 ,v1 ,v2] = EarthJupiterPosVel(Maxtdep,Maxtarr - Maxtdep);
% v1 = (v1*10 + r1);
% v2 = (v2*10 + r2);

% MaxV1 = (MaxV1*86400/kmAU + r1);
% plot3(r1(1),r1(2),r1(3),'bo')
% hold on
% plot3(r2(1),r2(2),r2(3),'yo')
% plot3([0 r1(1)],[0 r1(2)],[0 r1(3)],'k')
% plot3([0 r2(1)],[0 r2(2)],[0 r2(3)],'k')
% plot3([r1(1) v1(1)],[r1(2) v1(2)],[r1(3) v1(3)],'r')
% plot3([r2(1) v2(1)],[r2(2) v2(2)],[r2(3) v2(3)],'r')
% plot3([r1(1) MaxV1(1)],[r1(2) MaxV1(2)],[r1(3) MaxV1(3)],'r')
```

```
% % axis equal
% hold off
%
% fprintf(’With departure date of %s and arrival date of %s \n’, Maxtdep_char, Maxtarr_char)
% fprintf(’The space ship will require a launch velocity of %f \n’,norm(MaxV1))
% fprintf(’And will leave Jupiter at a velocity of %f \n’,MaxV);
%% Plot the contour
%lets do 3 ticks on the departure axis
xticks = linspace(tdep(1) ,tdep(end) ,15);
xticklabels = cell(size(xticks ));
for j = 1: length(xticks)
xticklabels{j} = char(datetime(xticks(j),...
’ConvertFrom’,’JD’,’Format’,’dd -MMM’));
end
%and 5 on the y axis
yticks = linspace(tarr(1) ,tarr(end) ,15);
yticklabels = cell(size(yticks ));
for j = 1: length(yticks)
yticklabels{j} = char(datetime(yticks(j),...
’ConvertFrom’,’JD’,’Format’,’dd -MMM’));
end
%define contour levels to plot
minlvl = min(min(deltavs));
maxlvl = max(max(deltavs));
lvls = round(linspace(ceil(minlvl),floor(maxlvl),10));
lvls = [round(minlvl ,1),lvls,round(maxlvl ,1)];
figure (1)
clf
[C1 ,h1] = contour(tdep ,tarr ,deltavs.’,lvls,’Linewidth’,2);
clabel(C1,h1 ,’Color’,’b’,’FontSize’,16)
% caxis([min(lvls),max(lvls )])
hold on
% contour(tdep ,tarr ,dts ,’k--’,’ShowText’,’on’)
hold off
set(gca ,’FontName’,’Times’,’FontSize’,16,’XTick’,xticks ,...
’XTickLabel’,xticklabels ,’YTick’,yticks ,’YTickLabel’,yticklabels)
xlabel(’V in’)
ylabel(’V out’)
title(’Jupiter Assist $\Delta$v (km s$^{-1}$)’,’Interpreter’,’LaTeX’)
set(gca ,’FontName’,’Times’,’FontSize’,16,’XTick’,xticks ,...
’XTickLabel’,xticklabels ,’YTick’,yticks ,’YTickLabel’,yticklabels)

save(’JupiterAssist_ContourData.mat’,’deltavs’,’finalV’);

function [R] = Rz(phi)
R = [cos(phi),sin(phi),0;
     -sin(phi),cos(phi),0;
     0,0,1];
end
```

A.4 Transfer Parameters Calculator

Transfer_Parameters.m: Script for calculating transfer parameters.

```
%Transfer parameter calculations
clear;clc;
kmAU = 149597870.700; %1 AU in km
%% Earth 2 Jupiter
load('Earth2Jupiter_ContourData.mat')

a_E2J = zeros(length(tdep),length(tarr));
e_E2J = zeros(length(tdep),length(tarr));
l_E2J = zeros(length(tdep),length(tarr));
for j = 1: length(tdep)
    for k = 1: length(tarr)
        r1(1) = MRE(j,k,1); r1(2) = MRE(j,k,2); r1(3) = MRE(j,k,3);
        r2(1) = MRJ(j,k,1); r2(2) = MRJ(j,k,2); r2(3) = MRJ(j,k,3);
        r1 = r1*kmAU;
        r2 = r2*kmAU;

        a_E2J(j,k) = (norm(r1) - norm(r2))/2;
        e_E2J(j,k) = norm(r2-r1)/(norm(r1) - norm(r2));

        CosTheta = dot(r1,r2)/(norm(r1)*norm(r2));
        dnue = acos(CosTheta);

        l_E2J(j,k) = (norm(r1)*norm(r2)/norm(r2-r1))*(1-cos(dnue));
    end
end

%% Jupiter Assist
load('JupiterAssist_ContourData.mat')

phi_JA = zeros(length(tdep),length(tarr));
e_JA = zeros(length(tdep),length(tarr));
a_JA = zeros(length(tdep),length(tarr));
rp_JA = zeros(length(tdep),length(tarr));

ehat = [1; 0; 0];
mu_j = 1.2668e8;
for j = 1: length(tdep)
    for k = 1: length(tarr)
        V1(1) = MV1(j,k,1); V1(2) = MV1(j,k,2); V1(3) = MV1(j,k,3);
        costheta = dot(ehat,-V1)/(norm(ehat)*norm(V1));

        a_JA(j,k) = -mu_j/(norm(V1)^2);
        e_JA(j,k) = -1/costheta;
        rp_JA(j,k) = a_JA(j,k) - a_JA(j,k)*e_JA(j,k);
        phi_JA(j,k) =  2*asin((1+rp_JA(j,k)*norm(V1)^2/mu_j)^-1);
    end
end

%% Jupiter 9 Planet 9
%departures can be between 21/12/2023 adn 30/06/2026
tdep0 = juliandate(datetime('2023-12-21')) + 1230;
tdepf = juliandate(datetime('2026-06-30')) + 1230;
```

```
tdep_2 = tdep0:1:tdepf;
%arrivals are between 80 days after the first departure and 120 days after
%the final departure
tarr0 = tdep0 + 5410;
tarrf = tdepf + 5410;
tarr_2 = tarr0:1:tarrf;

load('Jupiter2Planet9_ContourData.mat')

a_J2P9 = zeros(length(tdep_2),length(tarr_2));
e_J2P9 = zeros(length(tdep_2),length(tarr_2));
l_J2P9 = zeros(length(tdep_2),length(tarr_2));
for j = 1: length(tdep_2)
    for k = 1: length(tarr_2)
        r1(1) = MRJ(j,k,1); r1(2) = MRJ(j,k,2); r1(3) = MRJ(j,k,3);
        r2(1) = MRP9(j,k,1); r2(2) = MRP9(j,k,2); r2(3) = MRP9(j,k,3);
        r1 = r1*kmAU;
        r2 = r2*kmAU;

        a_J2P9(j,k) = (norm(r1) - norm(r2))/2;
        e_J2P9(j,k) = norm(r2-r1)/(norm(r1) - norm(r2));

        CosTheta = dot(r1,r2)/(norm(r1)*norm(r2));
        dnue = acos(CosTheta);

        l_J2P9(j,k) = (norm(r1)*norm(r2)/norm(r2-r1))*(1-cos(dnue));
    end
end

%% Analysis

normVnew = zeros(length(tdep),length(tarr));
for j = 1: length(tdep)
    for k = 1: length(tarr)
        Vnew(1) = finalV(j,k,1); Vnew(2) = finalV(j,k,2); Vnew(3) = finalV(j,k,3);
        normVnew(j,k) = norm(Vnew);
    end
end


MaxV1_2 = 0;
for j = 1: length(tdep)
    for k = 1: length(tarr)
        temp = normVnew(j,k);
        if (temp > MaxV1_2) && (rp_JA(j,k) > 70000.513)
            MaxV1_2 = temp;
            maxj = j;
            maxk = k;
        end
    end
end

V_diff = zeros(length(tdep),length(tarr));
for j = 1: length(tdep_2)
    for k = 1: length(tarr_2)
        V1_2(1) = MV1_2(j,k,1); V1_2(2) = MV1_2(j,k,2); V1_2(3) = MV1_2(j,k,3);
        V_diff = abs(norm(V1_2)- MaxV1_2);
```

```
    end
end

[M,I1] = min(V_diff);
[MinV_diff,I2] = min(M);
maxj_2 = I1(I2);
maxk_2 = I2;

V1(1) = MV1(maxj,maxk,1); V1(2) = MV1(maxj,maxk,2); V1(3) = MV1(maxj,maxk,3);
V1_2(1) = MV1(maxj_2,maxk_2,1); V1_2(2) = MV1(maxj_2,maxk_2,2); V1_2(3) = MV1(maxj_2,maxk_2,3);

Maxtarr_char = char(datetime(tarr(maxj),'ConvertFrom','JD','Format','dd/MMM/yyyy'));
Maxtdep_char = char(datetime(tdep(maxk),'ConvertFrom','JD','Format','dd/MMM/yyyy'));
Maxtarr_2_char = char(datetime(tarr_2(maxj_2),'ConvertFrom','JD','Format','dd/MMM/yyyy'));
Maxtdep_2_char = char(datetime(tdep_2(maxk_2),'ConvertFrom','JD','Format','dd/MMM/yyyy'));

fprintf('First Semi-major axis of %s \n', a_E2J(maxj,maxk))
fprintf('First Eccentricity of %f \n', e_E2J(maxj,maxk))
fprintf('First Semi-parameter of %f \n', l_E2J(maxj,maxk));
fprintf('First Transfer velocity %f \n', norm(V1));
fprintf('\n');
fprintf('Assist Semi-major axis of %s \n', a_JA(maxj,maxk))
fprintf('Assist Eccentricity of %f \n', e_JA(maxj,maxk))
fprintf('Assist Closest Approach of %f \n', rp_JA(maxj,maxk));
fprintf('Assist Turning Angle %f \n', phi_JA(maxj,maxk));
fprintf('\n');
fprintf('Second Semi-major axis of %s \n', a_J2P9(maxj_2,maxk_2))
fprintf('Second Eccentricity of %f \n', e_J2P9(maxj_2,maxk_2))
fprintf('Second Semi-parameter of %f \n', l_J2P9(maxj_2,maxk_2));
fprintf('Second Transfer velocity %f \n', norm(V1_2));
fprintf('\n');
fprintf('With departure date of %s and arrival date at Jupiter of %s \n', Maxtdep_char,
    Maxtarr_char);
fprintf('and arrival date at Planet 9 of %s \n', Maxtarr_2_char)
```

A.5 Earth to Jupiter Position Calculator

EarthJupiterPosVel.m: Script for calculating planetary positions.

```
function [r_e,r_v,v_e,v_v] = EarthJupiterPosVel(t0,dt)

%INPUTS
%   t0 - Initial time (JD)
%   dt - Cruise duration (days)
%
%OUTPUTS
%   r_e - 3x1 col vector - Heliocentric position of Earth (in AU) at t0
%   r_v - 3x1 col vector - Heliocentric position of Venus (in AU) at t0+dt
%   v_e - 3x1 col vector - Heliocentric-Inertial velocity of Earth
%                          (in AU/day) at t0
%   v_v - 3x1 col vector - Heliocentric-Inertial velocity of Venus
%                          (in AU/day) at t0+dt
```

```
%Earth and Venus orbital elements
a_e = 0.999987495;                  %sma (AU)
a_j = 5.20336301;
e_e = 0.0167;                       %eccentricity
e_j = 0.04839266;
I_e = 2.777040607882003E-03*pi/180; %Inclination (rad)
I_j = 1.30530*pi/180;
w_e = 3.043573249748720E+02*pi/180; %arg. of periapsis (rad)
w_j = 274.1977*pi/180;
O_e = 1.596967974767415E+02*pi/180; %long. of ascending node (rad)
O_j = 100.55615*pi/180;
t_p_e = 2458853.731945450883;       % time of periapsis passage (JD)
t_p_j = juliandate(datetime('2011-03-13'));
mu_e = 2.9591309705483544E-04;      %G(m_sun + m_planet) (AU^3/day^2)
mu_j = 2.9626e-04;

%add any subfunctions as needed.  remember to put an 'end' statement at the end of any subfunctions
    function [r,v] = IHelpFunction(mu,a,e,I,w,O,tp,t)
        n = sqrt(mu/a^3);       %mean motion
        %calculate the Mean anomaly at each of the values in t
        M = n*(t - tp); %mean anomaly
        E = Newton_Raphson(e,M);
        b = a*sqrt(1 - e^2);
        x = a*(cos(E) - e);
        y = b*sin(E);
        Ed = n/(1 - e*cos(E));
        vx = -a*Ed.*sin(E);
        vy = b*Ed.*cos(E);
        r_per = [x,y,0];
        v_per = [vx,vy,0];
        [r,v] = IHelpFunction2(r_per,v_per,I,w,O);
    end

    function [r,v] = IHelpFunction2(r_per,v_per,I,w,O)
        R1 = [cos(O) sin(O) 0;
              -sin(O) cos(O) 0;
              0 0 1];
        R2 = [1 0 0;
              0 cos(I) sin(I);
              0 -sin(I) cos(I)];
        R3 = [cos(w) sin(w) 0;
              -sin(w) cos(w) 0;
              0 0 1];
        r = transpose(R3*R2*R1)*r_per';
        v = transpose(R3*R2*R1)*v_per';
    end

    function [E] = Newton_Raphson(e,M)
        tol = 1e-13;
        if (M/(1-e)) < sqrt(6*(1-e)/e)
            E0 = M/(1-e);
        else
            E0 = (6*M/e)^(1/3);
        end
        Enew = E0;
        whileend = abs(M-(Enew-e*sin(Enew)));
```

```
        while whileend > tol
            Eold = Enew;
            Enew = Eold - (M - Eold + e*sin(Eold))/(e*cos(Eold) - 1);
            whileend = abs(M-(Enew-e*sin(Enew)));
        end
        E=Enew;
    end

%calculate position and velocity vectors
[r_e,v_e] = IHelpFunction(mu_e,a_e,e_e,I_e,w_e,O_e,t_p_e,t0);
[r_v,v_v] = IHelpFunction(mu_j,a_j,e_j,I_j,w_j,O_j,t_p_j,t0+dt);

%ensure outputs are column vectors
r_e = r_e(:);
r_v = r_v(:);
v_e = v_e(:);
v_v = v_v(:);
end
```

A.6 Earth to Planet 9 Position Calculator

JupiterPlanet9PosVel.m: Script for calculating planetary positions.

```
function [r_e,r_v,v_e,v_v] = JupiterPlanet9PosVel(t0,dt)

%INPUTS
%   t0 - Initial time (JD)
%   dt - Cruise duration (days)
%
%OUTPUTS
%   r_e - 3x1 col vector - Heliocentric position of Earth (in AU) at t0
%   r_v - 3x1 col vector - Heliocentric position of Venus (in AU) at t0+dt
%   v_e - 3x1 col vector - Heliocentric-Inertial velocity of Earth
%                          (in AU/day) at t0
%   v_v - 3x1 col vector - Heliocentric-Inertial velocity of Venus
%                          (in AU/day) at t0+dt

%Earth and Venus orbital elements
a_j = 5.20336301;                  %sma (AU)
a_9 = 380;
e_j = 0.04839266;                          %eccentricity
e_9 = 0.2105;
I_j = 1.30530*pi/180; %Inclination (rad)
I_9 = 16*pi/180;
w_j = 274.1977*pi/180; %arg. of periapsis (rad)
w_9 = 250*pi/180;
O_j = 100.55615*pi/180; %long. of ascending node (rad)
O_9 = 100*pi/180;
t_p_j = juliandate(datetime('2011-03-13'));       % time of periapsis passage (JD)
t_p_9 = 2458029.748282369226+790000;
mu_j = 2.9626e-04;      %G(m_sun + m_planet) (AU^3/day^2)
mu_9 = 1.488e-34*(1.9891e30 + 6.2*5.97219e24);
```

```

%add any subfunctions as needed.  remember to put an 'end' statement at the end of any subfunctions
    function [r,v] = IHelpFunction(mu,a,e,I,w,O,tp,t)
        n = sqrt(mu/a^3);       %mean motion
        %calculate the Mean anomaly at each of the values in t
        M = n*(t - tp); %mean anomaly
        E = Newton_Raphson(e,M);
        b = a*sqrt(1 - e^2);
        x = a*(cos(E) - e);
        y = b*sin(E);
        Ed = n/(1 - e*cos(E));
        vx = -a*Ed.*sin(E);
        vy = b*Ed.*cos(E);
        r_per = [x,y,0];
        v_per = [vx,vy,0];
        [r,v] = IHelpFunction2(r_per,v_per,I,w,O);
    end

    function [r,v] = IHelpFunction2(r_per,v_per,I,w,O)
        R1 = [cos(O) sin(O) 0;
              -sin(O) cos(O) 0;
              0 0 1];
        R2 = [1 0 0;
              0 cos(I) sin(I);
              0 -sin(I) cos(I)];
        R3 = [cos(w) sin(w) 0;
              -sin(w) cos(w) 0;
              0 0 1];
        r = transpose(R3*R2*R1)*r_per';
        v = transpose(R3*R2*R1)*v_per';
    end

    function [E] = Newton_Raphson(e,M)
        tol = 1e-14;
        if (M/(1-e)) < sqrt(6*(1-e)/e)
            E0 = M/(1-e);
        else
            E0 = (6*M/e)^(1/3);
        end
        Enew = E0;
        whileend = abs(M-(Enew-e*sin(Enew)));
        while whileend > tol
            Eold = Enew;
            Enew = Eold - (M - Eold + e*sin(Eold))/(e*cos(Eold) - 1);
            whileend = abs(M-(Enew-e*sin(Enew)));
        end
        E=Enew;
    end

%calculate position and velocity vectors
[r_e,v_e] = IHelpFunction(mu_j,a_j,e_j,I_j,w_j,O_j,t_p_j,t0);
[r_v,v_v] = IHelpFunction(mu_9,a_9,e_9,I_9,w_9,O_9,t_p_9,t0+dt);

%ensure outputs are column vectors
r_e = r_e(:);
r_v = r_v(:);
v_e = v_e(:);
```

```
v_v = v_v(:);
end
```

A.7 Solar Radiation Pressure Perturbations Script

SRP_perturbations.m: Script for estimating effects of solar radiation pressure on orbital elements.

```
clear
clc

%Earth and Planet 9 orbital elements
a_e = 1.000373836656026E+00;        %sma (AU)
a_9 = 520;
e_e = 1.712127710968187E-02;        %eccentricity
e_9 = 0.2596;
I_e = 2.777040607882003E-03*pi/180; %Inclination (rad)
I_9 = deg2rad(21);
w_e = 3.043573249748720E+02*pi/180; %arg. of periapsis (rad)
w_9 = deg2rad(250);
O_e = 1.596967974767415E+02*pi/180; %long. of ascending node (rad)
O_9 = deg2rad(100);
t_p_e = 2458853.731945450883;       % time of periapsis passage (JD)
t_p_9 = 2458029.748282369226;
mu_e = 2.9591309705483544E-04;      %G(m_sun + m_planet) (AU^3/day^2)
mu_9 = 1.488e-34*(1.9891e30 + 6.2*5.97219e24);
mass_e = 5.9722e24; %kg
mass_sun = 1.989e30; %kg
G = 6.6743e-11; % gravitational constant
mue_sun = G*(mass_e + mass_sun);
transferTime_earthtojupiter = 1227*24*60*60; % JD to seconds
transferTime_jupitertop9 = 5410*24*60*60; % JD to seconds
dt = transferTime_earthtojupiter + transferTime_jupitertop9; % estimated orbit time

Hs = 64e6; % sun surface radiation, W/m^2
Rs = 695e6; % sun radius, m
c = 2.998e8; % speed of light
A = 2; % approxite satellite normal surface area, m^2

% estimate position and velocity vectors of earth and p9
[~,xe,ye,vxe,vye] = kepler_2body(a_e,e_e,mue_sun);
[~,x9,y9,vx9,vy9] = kepler_2body(a_9,e_9,mue_sun);

% convert pos/vel vectors to inertial frame from perifocal
r0 = omegaIOmega_rot([xe(1);ye(1);0],O_e,I_e,w_e,1); % earth orbit radius
v0 = omegaIOmega_rot([vxe(1);vye(1);0],O_e,I_e,w_e,1); % earth orbit velocity
rf = omegaIOmega_rot([x9(1);y9(1);0],O_9,I_9,w_9,1); % planet 9 orbit radius
vf = omegaIOmega_rot([vx9(1);vy9(1);0],O_9,I_9,w_9,1); % planet 9 orbit velocity

% calculate average perturbation force over the full transfer
distances = linspace(norm(r0),norm(rf),10000);
Forces = 0;
for i = 1:length(distances)
```

```
    D = distances(i)*1.496e11; % convert au to m
    H = Rs^2/D^2*Hs;
    Forces = Forces + 2*H*A/c;
end

avgForce = Forces/length(distances);

% estimate average angular momentum
hAvg = (cross(r0,v0) + cross(rf,vf))/2;

% estimate rate change of eccentricity
dedt = avgForce*norm(hAvg)/mue_sun;

% estimate change in eccentricity over full transfer
de = dedt*dt;

fprintf('Estimated change in eccentricity due to solar radiation pressure is %g over the full Earth to Planet 9 traverse\n',de);

%% Functions

function [t,x,y,vx,vy,En] = kepler_2body(a,e,mu)
% Two-body problem using Newton-Raphson inversion of Kepler's
% time equation
% Inputs:
%   a - semi-major axis
%   e - eccentricity
%   mu - gravitational parameter (with same distance units as a)
%
% Output:
%   t - 1000x1 element time array between 0 and the orbital period
%   x,y - Orbital radius components for one orbit in the perifocal
%         frame (e,q) directions. Same size as t.
%   En - Specific orbital energy over one orbital period. Same
%       size as t.

tol = 2*eps(2*pi); %use this for the tolerance in your Newton-Raphson

b = sqrt(a^2*(1 - e^2));

n = sqrt(mu/a^3);       %mean motion
Tp = 2*pi*a^(3/2)/sqrt(mu);       %orbital period
h = 2*pi*a*b/Tp;       %angular momentum

%create time array from 0 to To with 1000 total points
t = linspace(0,Tp,1e3).';

%calculate the Mean anomaly at each of the values in t
% assume tp = 0

M = n*(t-0); %mean anomaly

%Newton-Raphson to find eccentric anomaly. Iterate to
%machine-percision tolerance (tol)

% set E array to be the right size
E = zeros(1,length(t));
```

```

looper = 0;

for i = 1:length(t)

    M1 = M(i);

    if M1/(1-e) < sqrt(6*(1-e)/e)
        E0 = M1/(1-e);
    else
        E0 = (6*M1/e)^(1/3);
    end


    if looper > 1000
        break
    end
    looper = 0;

    % iterate to solve for E until tolerance is met
    while(tol < abs(M1 - (E(i) - e*sin(E(i)))))

        abs(M1 - E(i) - e*sin(E(i)));

        looper = looper + 1;

        % dont get stuck in infinite loop
        if looper > 1000
            fprintf('infinite loop');
            break
        end

        % setup for fzero solver -- is fzero needed???
        myfun = @(Ei,M1,e) M1 - (Ei - e*sin(Ei));
        fun = @(Ei) myfun(Ei,M1,e);

        % compute current E(i) -- not sure if this is correct to do here
        Esolved = fzero(fun,E0);
        Enextguess = Esolved - (M1 - Esolved + e*sin(Esolved))/(e*cos(Esolved) - 1);
        E(i) = Esolved;
        E0 = Enextguess;

    end

end

%calculate x and y positions in the perifocal frame using Kepler's
%equations
x = a.*(cos(E)-e);
y = b.*sin(E);

%calcualte the x and y velocities in the perifocal frame

Edot = n./(1-e*cos(E));

vx = -a.*Edot.*sin(E);
vy = b.*Edot.*cos(E);
```

```

z = [x;y;vx;vy]';

%calculate the specific orbital energy
En = (z(:,3).^2+z(:,4).^2)/2-mu./sqrt(z(:,1).^2+z(:,2).^2); % E = (v^2)/2-mu/r

%force all outputs to be col vectors
x = x(:);
y = y(:);
En = En(:);
end

% function to convert between perifocal and inertial frames

function [vect, frm] = omegaIOmega_rot(vec,omega,I,w,dir)
%INPUTS
%   vec - Vector to be transformed
%   omega - Longitude of the ascending node [rad]
%   I - Inclination             [rad]
%   w - Argument of periapsis [rad]
%   dir - 0 for Inertial to Perifocal, 1 for Perifocal to Inertial

%OUTPUTS
%   vect - Transformed Vector
%   frm - Reference Frame of Vector

%Individual Direction Cosine matricies for 3-1-3 Rotation
dcOmega = [cos(omega) sin(omega) 0;
    -sin(omega) cos(omega) 0;
    0 0 1];
dcI = [1 0 0;
    0 cos(I) sin(I);
    0 -sin(I) cos(I)];
dcw = [cos(w) sin(w) 0;
    -sin(w) cos(w) 0;
    0 0 1];
%Composed Direction Cosine Matricies for frame transformations
pCi = dcw * dcI * dcOmega;
iCp = pCi.';

%Determine Direction
    if dir == 0
        vect = pCi * vec;
        frm = 'Perifocal';
    elseif dir == 1
        vect = iCp * vec;
        frm = 'Inertial';
    elseif dir ~= 0 && dir ~=1
        error('Direction of Transformation may be either 0 or 1')
    end
end
```

A.8 Solar Panel Requirements Script

---

solarPowerCalculations.m: Script for estimating required solar panel size to achieve given power outputs at Jupiter and Planet 9.

```
clear
clc

Hs = 64e6; % sun surface radiation, W/m^2
Rs = 695e6; % sun radius, m
D = 778e9; % jupiter distance from sun, m
maxD = 385*1.496e11; % p9 distance from sun, m

HJ = Rs^2/D^2*Hs; % power intensity at jupiter, w/m^2
H9 = Rs^2/maxD^2*Hs; % power intensity at planet 9, w/m^2

minPowerJ = 2300; % minimum power required for spacecraft operation
minPower9 = 1;
efficiency = .3; % assumed

panelAreaJ = minPowerJ/(HJ*efficiency);
panelArea9 = minPower9/(H9*efficiency);

fprintf('The required solar panel surface area is %.3f square meters for 2.3 kW of solar power at
    Jupiter.\n',panelAreaJ);
fprintf('\n');
fprintf('The required solar panel surface area is %.3f square meters for 1 W of solar power at
    Planet 9\n',panelArea9);
```

A.9 Mission Name

The High Velocity Interstellar and Deep Space Planet Reconnaissance Spacecraft or VIDAR is named for the Norse god Víðarr known as "The Silent God", an apt description of Planet 9. VIDAR will depart Earth's sphere of influence with the highest energy and velocity of any artificial object in human history, and will then fire its ion thrusters to continue accelerating. It will pass the Voyager probes, covering a distance that took them 50 years in 10. After its encounter with Planet 9, VIDAR will continue on at these incredible velocities reaching 1000 AU within 60 years of launch.