Comprehensive Matlab GUI for Determining Barycentric Orbital Trajectories

Steve Katzman[1]
California Polytechnic State University, San Luis Obispo, CA 93405

When a 3-body gravitational system is modeled using a rotating coordinate frame, interesting applications become apparent. This frame, otherwise known as a barycentric coordinate system, rotates about the system's center of mass. Five unique points known as Lagrange points rotate with the system and have numerous applications for spacecraft operations. The goal of the Matlab GUI was to allow easy manipulation of trajectories in a barycentric coordinate system to achieve one of two end goals: a free-return trajectory or a Lagrange point rendezvous. Through graphical user input and an iterative solver, the GUI is capable of calculating and optimizing both of these trajectory types for all of our solar system's planets. Its inputs are inertial state vectors, a date and time, and the number of propagation days. The user can then graphically manipulate the resulting trajectories by increasing the spacecraft velocity and propagation start time. It outputs the resulting ΔV vectors and magnitudes as well as a graphical representation of the desired orbital path.

Nomenclature

r	=	radius used in barycentric orbital propagation
x	=	barycentric x-axis radius value
y	=	barycentric y-axis radius value
z	=	barycentric z-axis radius value
μ^*	=	mass ratio of the barycentric system

Subscripts

0	=	initial state

I. Introduction

THE barycentric coordinate system utilized in this project operates under a specific set of assumptions called the circular restricted three-body problem or CR3BP. First, it forces each of the two gravitational bodies into a circular orbit. These circular orbits rotate around the center of mass of the system, otherwise known as the barycenter. Second, the model is restricted to only two gravitational bodies and a third orbiting body of negligible mass. Thus, n-body effects from other planets or the sun are always neglected. Additionally, all perturbing effects due to solar radiation, drag or oblateness are also ignored.

Five specific points in space orbit with the rotating frame. These points in space are called Lagrange points and each one has unique properties useful for spacecraft operations. An image of the Lagrange points for the Earth-Moon system can be seen in Figure 1. L1, L2 and L3 all lie in-line with the primary and secondary bodies' orbits. L2 is between the two and can be most easily understood as the point in space where the two bodies' gravities cancel out. L1 is on the other side of the secondary body and represents the point where the added effects of the two bodies' gravities achieve a centripetal acceleration that causes the point to rotate about the barycenter at the same rate as the coordinate frame. Similarly, L3 is on the opposite side of the primary as the secondary body and has just enough centripetal acceleration to rotate perfectly with the barycentric frame. L4 and L5 are points whose positions make equilateral triangles with the primary and secondary bodies off of the system's x-axis. These are the most stable

[1]Aerospace Undergraduate, Aerospace Department, 1 Grand Ave.

Lagrange points and it is possible to place a spacecraft at one of these two points such that it seems stationary with respect to the coordinate system and both gravitational bodies.

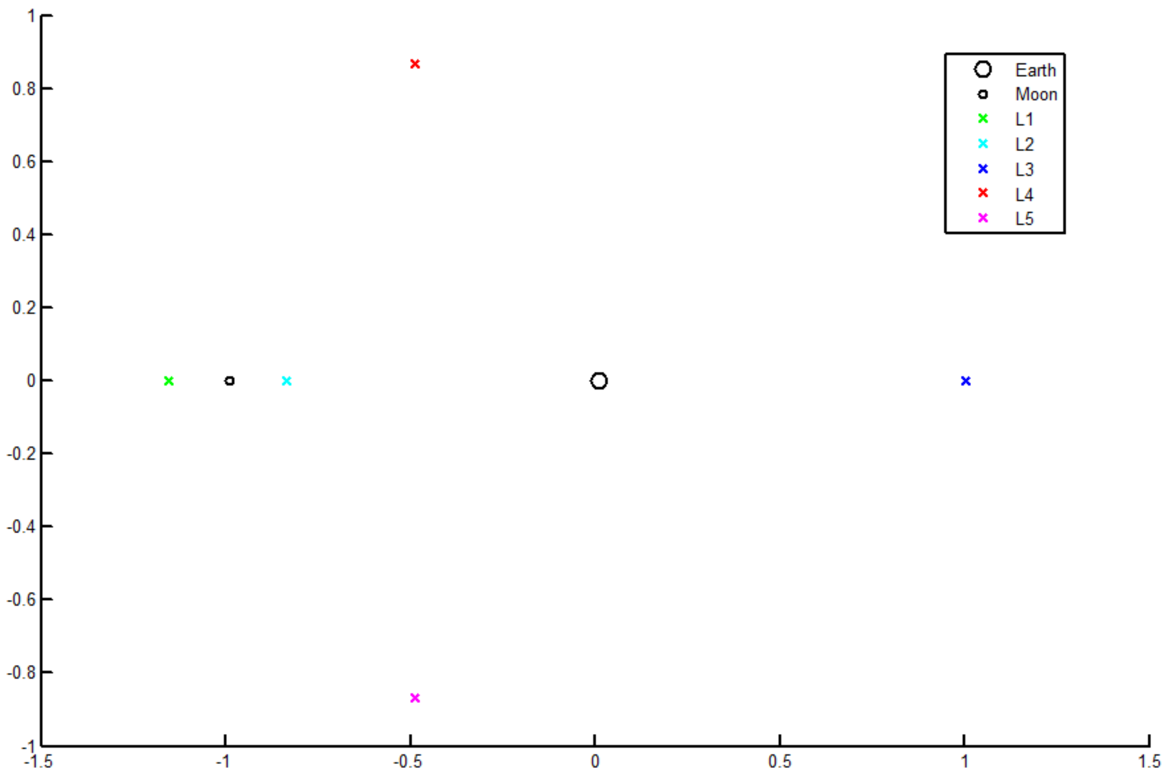


Fig. 1 **A depiction of the Lagrange points for the Earth-Moon system. Axes are in non-dimensional canonical distance units. The naming convention here is consistent throughout this project, however some references switch the naming conventions of L1, L2 and L3.**

Using the unique properties of Lagrange points, some interesting spacecraft operations become available. For example, a spacecraft located at the L1 point of the Earth-Moon system could act as a communications relay for dark-side, moon surface-based operations. The James Webb telescope will be positioned at the Sun-Earth L1 point such that the spacecraft is mostly shaded from the sun by Earth at all times. The Trojan asteroids are interestingly located at the Sun-Jupiter L4 and L5 points due to these points' high stability. In order to utilize these points, a spacecraft must first rendezvous with the Lagrange point. Thus, one of the goals of this project was to allow spacecraft maneuvering to any of the five Lagrange points.

Another property of the CR3BP system is the ability to accomplish free-return trajectories. This is an orbital path which takes the spacecraft to the secondary body and then allows it to return to the primary without any additional fuel. This class of orbit was used for the manned-missions to the moon during the Apollo 8, 10 and 11 missions. The manned capsule was launched onto a free-return trajectory to ensure it could return home if the propulsion system checks failed. Though all three of these missions were successful and did not require the use of the free-return, Apollo 13 utilized one to return home. This mission was capable of moving itself onto a free-return trajectory once the system checks determined spacecraft malfunctions and the astronauts were delivered safely home. Determining these useful orbits was another goal of this project.

II. Analysis

The input state vectors to the GUI are in the inertial frame of primary. In order to perform analysis in the rotating barycentric frame, a number of rotations were applied to the spacecraft state vectors to place them into this frame. First the secondary body is propagated to its inertial position based on the user input date and time. Then a rotation into its perifocal frame is accomplished. The coordinate transform needed to accomplish this rotation is saved as a global variable within the GUI. The second rotation places the secondary body on the x-axis to the left of the primary. This is a simple rotation about the z-axis defined by the difference in 180° and the secondary's true anomaly. Like the first coordinate transformation, the 3x3 rotation matrix is saved as a global variable.

These two rotations are applied to the spacecraft's inertial vectors, placing it into the time-dependent barycentric coordinate system. Additionally, the rotation of the frame is accounted for in the spacecraft's barycentric velocity. The entire system is then converted into canonical units. The distance unit is defined as the semi-major axis of the secondary body. The time unit is defined such that the standard gravitational parameter, μ, is 1. The same conversions used at the beginning of the calculations are performed in reverse at the end of the operations such that the GUI outputs are once again the primary's inertial frame.

The initial canonical barycentric state vectors are fed into an algorithm which performs CR3BP propagation. The equations of motion used in the numerical integration are shown below:

$$\ddot{x} = 2\dot{y} + x - \frac{(1-\mu^*)(x-\mu^*)}{r_1^3} - \frac{\mu^*(x+1-\mu^*)}{r_2^3} \qquad \text{(Eq. 1)}$$

$$\ddot{y} = -2\dot{x} + y - \frac{(1-\mu^*)y}{r_1^3} - \frac{\mu^* y}{r_2^3} \qquad \text{(Eq. 2)}$$

$$\ddot{z} = -\frac{(1-\mu^*)z}{r_1^3} - \frac{\mu^* z}{r_2^3} \qquad \text{(Eq. 3)}$$

where μ^* is the mass ratio of the barycentric system, *x, y,* and *z* represent the barycentric coordinates, and r_1 and r_2 are defined below:

$$r_1 = \sqrt{(x-\mu^*)^2 + y^2 + z^2} \qquad \text{(Eq. 4)}$$

$$r_2 = \sqrt{(x-\mu^*+1)^2 + y^2 + z^2}. \qquad \text{(Eq. 5)}$$

To accomplish a free-return trajectory or Lagrange-point rendezvous it was determined that the spacecraft's velocity needed to be restricted to the barycentric xy-plane. This mitigated out of plane propagation and reduced a degree of freedom for the spacecraft when iteratively solving for the desired orbit. Thus, an inclination change was applied to the spacecraft inertial vectors before moving into the barycentric frame when performing orbital determination calculations beyond the initial propagation. Initial COEs for the secondary body's and spacecraft's orbits were determined and then an inclination change algorithm was employed to move the spacecraft into the barycentric xy-plane. This generated the initial vectors for the user to graphically manipulate, discussed in greater detail in Section III.

To solve a free-return trajectory, an iterative solving algorithm was implemented. A range of ΔV values and velocity directions were applied to the spacecraft and the resulting state vectors were then propagated. Radius vectors were checked for the following success conditions:

1. The spacecraft must be within 8% of the secondary body's x-position.
2. This must occur at an x-axis crossing.

Once a successful orbit was determined, the algorithm re-runs with higher fidelity spans. One of the new span limits the ΔV such that a "refined trajectory" is never greater than its "course" counterpart. Another limits the velocity direction to near the "course" solution's to minimize the number of propagations needed to run. An initial span of 200 ΔV values and 100 angles of rotation are applied to determine the "course trajectory".

The outputs of this algorithm are initial state vectors for final barycentric propagation and the ΔV vectors needed to accomplish it including inclination change and the burn needed to put the spacecraft onto a free-return trajectory.

To accomplish a Lagrange point rendezvous, a very similar algorithm is employed. An inclination change is first applied to spacecraft and then the canonical barycentric state vectors are iterated upon until the specified Lagrange point is reached. For this algorithm, more precise spans for ΔV were included. Additionally, the tolerances on the success conditions were tightened to ensure the spacecraft actually intercepts the Lagrange point. The specific success conditions are:

1. The spacecraft must be within 1% of the specified Lagrange point's position in both the x and y coordinates.
2. The orbit is accomplished with less than the maximum allotted ΔV.

A similar refining algorithm is run to optimize the spacecraft's fuel consumption. Because of the increased span sizes and tolerances, determining a Lagrange point rendezvous takes significantly more time than finding a free-return trajectory.

Once an orbital path has been determined via iterative method, its output state vectors are propagated and plotted. The ΔV vectors and magnitudes are converted out of the canonical barycentric frame and back into the inertial coordinate system. These values and vectors become the numerical results displayed in the GUI after optimization is complete.

III. Matlab GUI Operation

As stated above, the GUI takes in inertial state vectors, a date and time, and a propagation time in days. The user also specifies a primary and secondary body. This accommodates systems with the Sun as the primary or any of the planets as a primary with their moon(s) as a secondary. The GUI is pictured before running in Figure 2.

After declaring these initial parameters, the spacecraft can be propagated without any changes applied by pressing the "Plot Initial Orbit" button. Not only does this button provide three-dimensional results of the propagation but also defines a number of parameters used in later calculations. The GUI is shown in Figure 3 after running the "Plot Initial Orbit" button and before specifying/optimizing a trajectory option.

At this point, it is then up to the user to specify the desired trajectory. If the user selects "L-Point Rendezvous" from the Trajectory Option's listbox, five radio buttons corresponding to the five Lagrange points become visible. This panel of the GUI also includes a "Guess Trajectory" sub-panel. The "Increase Velocity" bar linearly increases the spacecraft's velocity in its current initial velocity direction. For planet-moon systems, the maximum increase in velocity is 5 km/s. For sun-planet systems, this maximum value is increased to 15 km/s. As soon as the slider is manipulated, the resulting orbit is propagated and displayed in the graphical results.

The user also has the option to increase the start date by sliding the "Start Date" slider. Like the velocity slider, any changes in the slider position are instantly propagated and displayed in the graphical results. This allows the user to better line up the desired trajectory before optimization is run. For planet-moon systems, the maximum increase in start date is 1 month. For sun-planet systems, this value is increased to 4 years.

Using these two sliders it is possible for the user to graphically design the desired trajectory. Once the graphical results match the desired end result, the trajectory is optimized by pressing the "Optimize It!" button. Pressing this button then runs the iterative algorithms discussed above. The ΔV and start date are taken from the slider values and used in these functions to greatly increase the accuracy of the spans used to iterate. For a free-return trajectory, the change in velocity specified by the slider is spanned through values of +/- 5% the slider value. For Lagrange point rendezvous this tolerance is increased to +/- 10% due to the increased sensitivity of these trajectories. This live propagation may run slowly on slower systems due to the intense computations required to propagate these orbits in real-time. However, on a fast enough system, the trajectories can seamlessly be manipulated in real time.

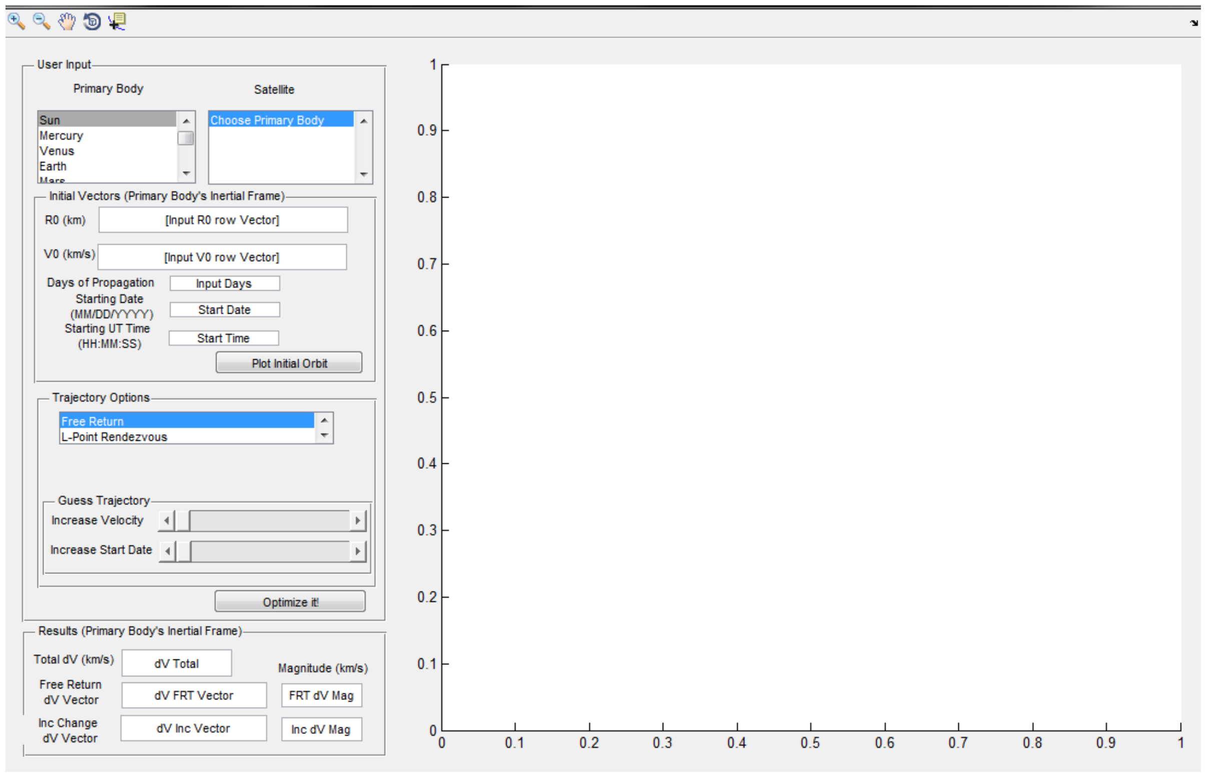


Figure 2 **The Lagrange_Point_GUI before running. The user can select a primary body (either the sun or a planet) and then any of the body's major satellites (including all the planets and major moons). The initial vectors, date and time and propagation time are also specified in the top panel. Once these initial parameters are defined, the user plots the initial orbit using the button in this panel. Then the user can specify the desired trajectory and manipulate the velocity and date sliders until a reasonable orbit is displayed in the plot. To optimize this orbit, the user can press the "Optimize It!" button which will display both graphical results in the plot and numerical results in the Results panel.**

The user-created trajectory needs to be very similar to what he/she wishes the output to be. This is because of the tight tolerances applied to the user-specified ΔV values. If the GUI's calculations cannot optimize the user-specified orbit, a text output in the command window will prompt him/her to better refine their inputs. If the desired trajectory does not seem feasible based on graphical user manipulation alone, it helps to fiddle with the specified start date and time since the propagation is time-dependent.

Once the optimize button has been pressed two wait bars will appear, one after the other. The first dictates the time until the coarse trajectory is achieved. This bar will rarely complete before closing and starting the second. The second specifies the time until the refined trajectory is completed. This bar should always complete. Once both wait bars close, the results of the optimization will be realized within the GUI both graphically and numerically.

IV. Results of Test Cases

The first simulation involves determining a free-return trajectory for the Earth-Moon system. The spacecraft is initialized in a geosynchronous orbit with the following inertial parameters:

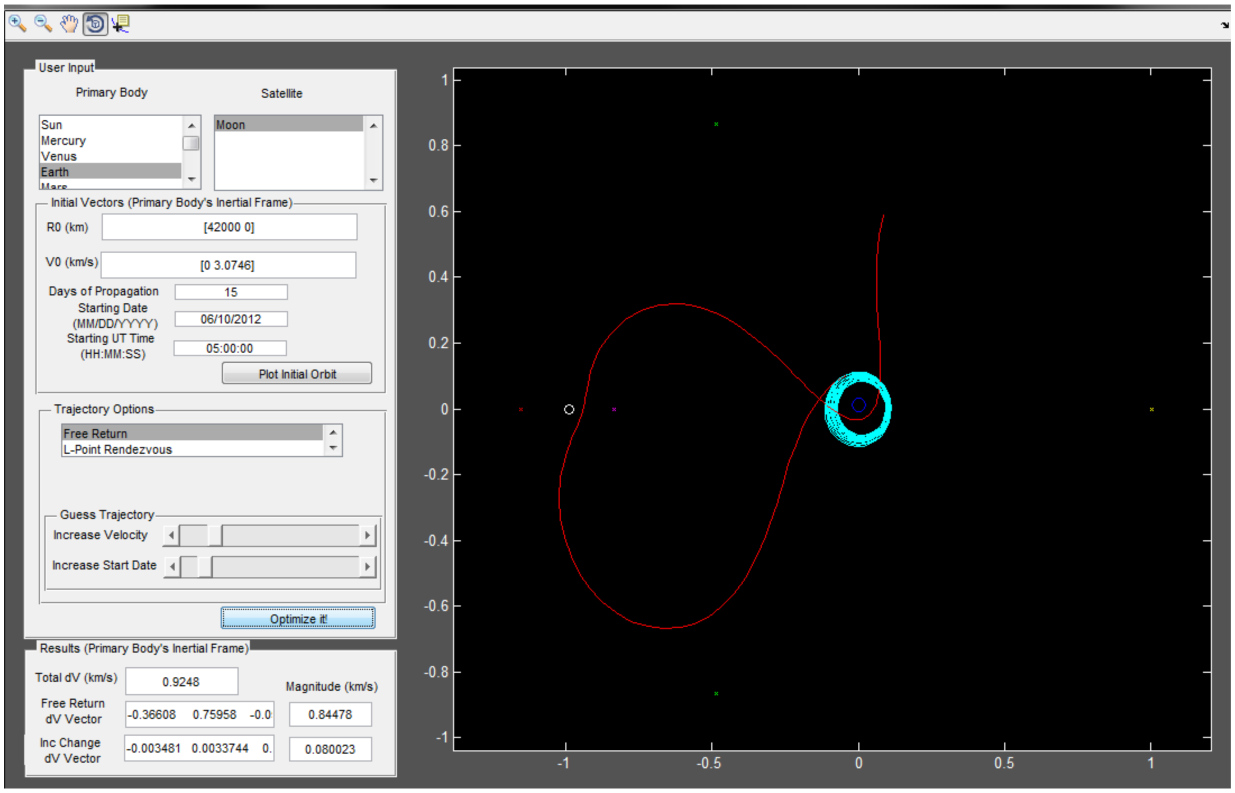
R_0 = [42000 0] (km)
V_0 = [0 3.0746] (km/s)
Starting date and time: 06/10/2011 at 05:00:00
Propagation time: 15 days

The results of this test cases' initial state is shown as the cyan orbit in Figure 3. The orbit seems to change shape (though it shouldn't) because of the low tolerance utilized within barycentric ODE solver which causes truncation error for higher order accelerations. However, the tolerances cannot be raised due to the fact that the GUI is operating in canonical units. After applying small changes in the sliders, seen in Figure 3, the orbit was optimized. As can be seen in the figure, the resulting orbit is a free return trajectory which does an Earth-side flyby of the moon and then returns to Earth without further fuel expenditure.

With the optimization techniques employed, the spacecraft requires an additional 925 m/s ΔV to accomplish both the inclination change and to place it onto its free-return trajectory. This is a very reasonable ΔV to achieve an orbit which reaches the moon from GEO. The full vector outputs from the GUI are shown below:

Free-return ΔV vector: [-0.366 0.760 -0.015] (km/s)
Inclination change vector: [-.003 0.003 0.079] (km/s)

As can be seen, the inclination change is very efficient due to the small change in inclination and altitude of the GEO orbit.



R0 (km) [42000 0]
V0 (km/s) [0 3.0746]
Days of Propagation 15
Starting Date (MM/DD/YYYY) 06/10/2012
Starting UT Time (HH:MM:SS) 05:00:00

Total dV (km/s) 0.9248
Magnitude (km/s)
Free Return dV Vector -0.36608 0.75958 -0.0 0.84478
Inc Change dV Vector -0.003481 0.0033744 0. 0.080023

Figure 3 **Screenshot of the Matlab GUI after running optimization on the specified Earth-Moon free-return trajectory. Vector displays at the bottom of the GUI are selectable text for easy integration into other numerical orbital analysis.**

The second simulation involves determining a Lagrange point rendezvous trajectory for the Earth-Moon system. The spacecraft is initialized in a similar geosynchronous orbit with the following inertial parameters:

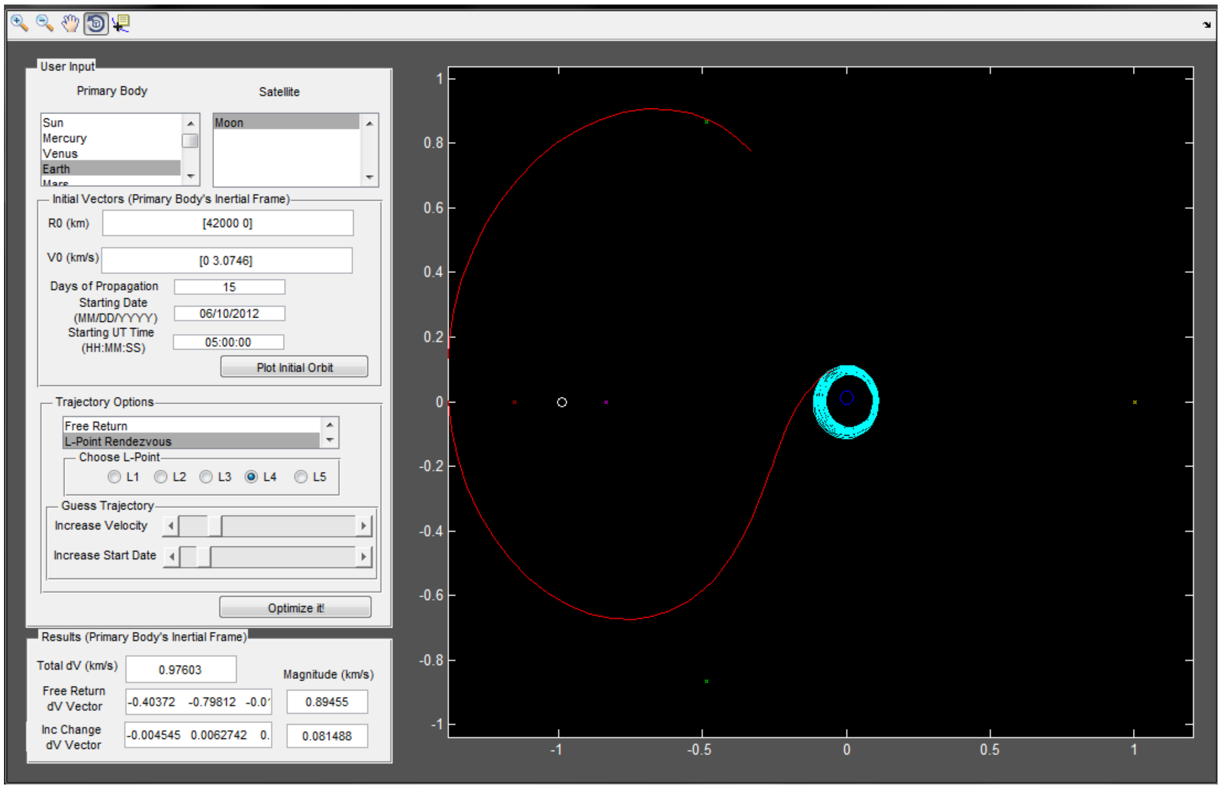
R_0 = [42000 0] (km)
V_0 = [0 3.0746] (km/s)
Starting date and time: 06/10/2011 at 05:00:00
Propagation time: 15 days

The results of this test cases' initial state is shown as the cyan orbit in Figure 4. After applying changes in the velocity and date sliders, seen in Figure 4, the orbit was optimized. As can be seen in the figure, the resulting orbit is a Lagrange point rendezvous to L4 which utilizes a high-altitude flyby of the moon and then continues on to the specified orbit without further fuel expenditure. The L1 through L5 radio buttons are visible in this test case below the trajectory listbox with L4 selected.

Once optimized, the spacecraft GUI provides that an additional 976 m/s ΔV to accomplish both the inclination change and to place it onto its rendezvous trajectory. This ΔV is higher than the first test cases' because of the increased energy needed to reach the outer Lagrange points (L4 and L5). This is a reasonable ΔV to achieve an orbit which reaches these further Lagrange points from GEO. The full vector outputs from the GUI are shown below:

Free-return ΔV vector: [-0.404 0.798 -0.021] (km/s)
Inclination change vector: [-.005 0.006 0.081] (km/s)

This type of trajectory is useful for capturing at a Lagrange point. In this test case specifically, the spacecraft arrives at the Lagrange point near its apoapse which means that a relatively low capture ΔV would be required to permanently place a spacecraft in this orbital regime.


User Input
Primary Body: Sun, Mercury, Venus, [✓ Earth], Mars
Satellite: [✓ Moon]

Initial Vectors (Primary Body's Inertial Frame)
R0 (km): [42000 0]
V0 (km/s): [0 3.0746]
Days of Propagation: 15
Starting Date (MM/DD/YYYY): 06/10/2012
Starting UT Time (HH:MM:SS): 05:00:00

Trajectory Options: Free Return, [✓ L-Point Rendezvous]
Choose L-Point: L1 L2 L3 [✓ L4] L5

Results (Primary Body's Inertial Frame)
Total dV (km/s): 0.97603
Free Return dV Vector: -0.40372 -0.79812 -0.0 — Magnitude (km/s): 0.89455
Inc Change dV Vector: -0.004545 0.0062742 0. — Magnitude (km/s): 0.081488

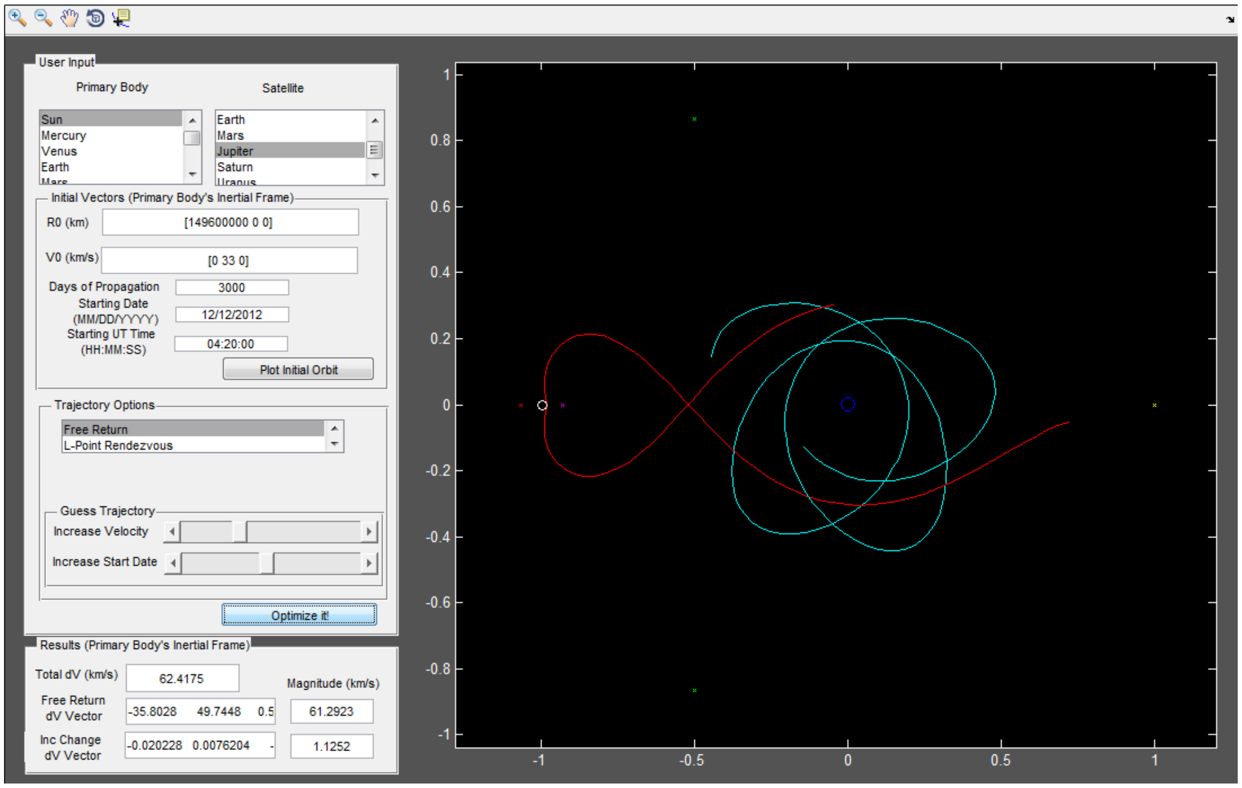
Figure 4 **Screenshot of the GUI after running the Lagrange point rendezvous optimization. The specified point, L4, has been reached with a reasonable fuel expenditure.**

The final test case involves a free-return trajectory from Earth to Jupiter in the Sun-Jupiter system. Unlike the previous two test cases, this simulation uses the increased max slider values and much higher propagation time. The GUI inputs used to run this test case follow:

R_0 = [149600000 0 0] (km)
V_0 = [0 33 0] (km/s)
Starting date and time: 12/12/2012 at 04:20:00
Propagation time: 3000 days

The initial orbit is plotted in cyan in Figure 5. Because the input parameters simulate launching from Earth with a C3 of roughly 15.5 km^2/s^2, the resulting orbit is eccentric and causes the propagation to look the way it does. After adjusting the sliders, the free-return trajectory was optimized. This utilized a significant increase in both ΔV and start date. As can be seen from the resulting solution orbit, it seems like the spacecraft leaves from an arbitrary point in space. This is because the orbit propagation between the start date and the adjusted start date is not shown on this plot. Further versions of this code should plot this propagation as well as the resulting start date, especially for sensitive systems like this interplanetary model.

The outputs of the GUI for this case are not as optimal as the previous two. The GUI provides a solution which states that a ΔV of 62.4 km/s is required to place the spacecraft on the shown free-return path. These incredibly high results are due a couple of factors. The first is that no inner planet flybys are utilized to help the trajectory get out to Jupiter. With this relatively low C3, inner planet flybys would always be utilized to reach Jupiter. Another factor is the GUI is not robustly designed to optimize the resulting ΔV vector. As in this case, the GUI calculated a trajectory which did not properly utilize the spacecraft's initial velocity and direction. Once again, further increases in optimization techniques would mitigate this blaring problem with the solution. Unfortunately, those explored were unsuccessful or incredibly inefficient, increasing the run time of the GUI to multiple hours. The final results of this run can be seen in Figure 5.



R0 (km): [149600000 0 0]
V0 (km/s): [0 33 0]
Days of Propagation: 3000
Starting Date (MM/DD/YYYY): 12/12/2012
Starting UT Time (HH:MM:SS): 04:20:00

		Magnitude (km/s)
Total dV (km/s)	62.4175	
Free Return dV Vector	-35.8028 49.7448 0.5	61.2923
Inc Change dV Vector	-0.020228 0.0076204 -	1.1252

Figure 5 **A Jovian free-return trajectory is displayed in this GUI screen capture. As can be seen a very high propagation time is necessary in order to ensure the resulting orbit is in fact a free-return. Also note that the scale of the planet specifier is not to scale and the trajectory does not take the spacecraft through the planet.**

V. Conclusion

The comprehensive Matlab GUI is capable of performing free-return trajectory analysis and Lagrange point rendezvous calculations. As was the goal, it takes in the spacecraft inertial vectors and is able to provide a barycentric graphical solution along with the inertial ΔV vectors needed to achieve the desired orbit. With user input, the GUI is capable of performing and optimizing these trajectories quickly and efficiently. By altering the velocity and start date of the spacecraft, it is possible for a user to graphically create a trajectory which seems close to the desired solution. The GUI's iterative solver is then able to achieve the orbit.

In its current state, this GUI is not perfect. The iterative solver methods used for orbital optimization could be overhauled to achieve "smarter" trajectories. Though the shapes of the resulting orbit are correct, as was seen in test case three, it sometimes chooses trajectories which are not optimal. For moon systems however, this GUI is quite robust and can efficiently calculate both forms of trajectories and optimize fairly well. Fortunately, free-return trajectories are more applicable in the Earth-Moon system because their greatest potential use is for manned missions.

A full running version of the code presented in this report should be available for free access through the California Polytechnic State University's Aerospace department. Because there are more than 5000 lines of code needed to run the GUI, the raw Matlab code is not included in the appendix.

References

Eagle, C. David. *The Circular-Restricted Three-Body Problem.* Orbital Mechanics with Matlab. 2012. http://www.cdeagle.com/ommatlab/crtbp.pdf.

Tuason, Christopher M. T. *Energy Potential Analysis of Zero Velocity Curves in the Restricted Three-Body Problem.* Dept. of the Air Force, Austin, TX, 1993.

Curtis, Howard. *Orbital Mechanics for Engineering Students.* Elsevier, Burlington, MA, 2010, pp. 129-145.

Vallado, David. *Fundamentals of Astrodynamics and Applications.* Space Technology Library, Hawthorn, CA, 2007, pp. 40-46.

Wheeler, Robin. "Apollo lunar landing launch window: The controlling factors and constraints". NASA, Retrieved 2009-10-27.

NASA Jet Propulsion Laboratory, "HORIZONS System," Pasadena, 2012.

Ceriotti, M. *Halo Orbit Determination in the Mission Analysis of the Hevelius-Lunar Microsatellite Mission.* University of Strathclyde, Glasgow, Scotland, 2005.