Automatic Generation of Low-Thrust Trajectories from LEO to Earth-Moon Lagrange Point Orbits

Andrew Abraham

11/16/2012

Abstract

This proposal will outline the goals of Andrew Abraham’s Ph.D. dissertation which focuses on topics found in Astrodynamics including low-thrust and three-body dynamics. After this document has been reviewed and accepted by both the Ph.D. Committee and the College of Engineering, Abraham will have fully satisfied all requirements to become a Ph.D. Candidate in Lehigh’s Department of Mechanical Engineering and Mechanics. Abraham will focus his research on the development of a rapid and robust algorithm that calculates a low-thrust, spiral trajectory from a geocentric orbit to an Earth-Moon Lagrange point orbit.

Contents

1 Proposal 2

1 Introduction 3

2 Doctoral Qualifications 3

2.1 Previous Education 3

2.2 Ph.D. Committee 4

2.3 Department of Mechanical Engineering & Mechanics Requirements 4

2.4 P.C. Rossin College of Engineering and Applied Science Requirements 5

3 State of the Art in Trajectory Design 5

3.1 Copernicus 5

3.2 MALTO 5

3.3 Mystic 5

3.4 OTIS 6

3.5 SNAP 6

3.6 CHEBYTOP 6

3.7 VARITOP/SEPTOP/NEWSEP 6

3.8 LTTool 6

4 Optimization Algorithms and Force Models 6

4.1 Coordinate System and Definitions 7

4.2 System Dynamics 8

4.3 State Transition Matrix 10

4.4 Generalized Free Variable & Constraint Vectors 12
Part I
Proposal

1 Introduction

Mission planners of today rely heavily on a suite of algorithms and software to adequately plan the trajectory of various spacecraft missions. Not all algorithms are suited for general use, however. Many algorithms are specifically designed to accomplish general tasks (i.e. Satellite Tool Kit). Unfortunately, many trajectory planning problems are too difficult to deal with using generalized software alone. A number of programs are currently available to address many specialized trajectory problems.

At present, no program nor algorithm currently exists that can rapidly and robustly calculate a low-thrust trajectory from a geocentric orbit (i.e. Low Earth Orbit) to an Earth-Moon Libration point orbit. This type of mission would be particularly useful for a low-thrust science or communications spacecraft intended for a Lagrange point orbit which cannot arrive by chemical means alone (due to mass constraints or a plethora of other reasons). The goal of this dissertation is to formulate a rapid and highly robust algorithm that can determine this type of low-thrust trajectory in a multi-body environment. This tool will merge long-duration, low-thrust, geocentric spirals with 3-body dynamics derived from Dynamical Systems Theory (DST). Inspiration for this work has come from Dr. Martin Ozimek and Dr. Chris Scott; two researchers at Johns Hopkins Applied Physics Laboratory (APL). Mr. Abraham has begun a research collaboration with them and will be receiving guidance from them on an intermittent basis.

This proposal will outline much of the work that has already been accomplished towards this goal, as well as planned work, and a proposed time-line for this work. A review of Mr. Abraham’s qualifications will be conducted as well as a thorough outline of the requisite knowledge needed to accomplish this task. This proposal concludes with an appendix that contains a lengthy review of background material that is relevant to three-body dynamics.

2 Doctoral Qualifications

The qualifications for a doctoral degree at Lehigh University are set at two levels: the department level and the college level.

2.1 Previous Education

<table>
<thead>
<tr>
<th>School</th>
<th>Degree</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>Moravian College</td>
<td>B.A. Economics</td>
<td>2009</td>
</tr>
<tr>
<td>Moravian College</td>
<td>B.S. Physics</td>
<td>2009</td>
</tr>
<tr>
<td>Lehigh University</td>
<td>M.S. Physics</td>
<td>2011</td>
</tr>
</tbody>
</table>

Table 1: Education

Mr. Abraham graduated from Moravian College Summa Cum Laude with Honors in Physics.
2.2 Ph.D. Committee

The Ph.D. Committee for Mr. Abraham’s doctoral degree has been established in Table 2:

<table>
<thead>
<tr>
<th>Name</th>
<th>School/Department</th>
<th>Position</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dr. Blythe</td>
<td>LU/Mechanical</td>
<td>Chair</td>
<td>Expert in engineering mathematics and thermofluids</td>
</tr>
<tr>
<td>Dr. Hart</td>
<td>LU/Mechanical</td>
<td>Adviser</td>
<td>Expert in spacecraft &amp; satellites; former Astronaut</td>
</tr>
<tr>
<td>Dr. Spencer</td>
<td>PSU/Aerospace</td>
<td>Outside Examiner</td>
<td>Expert in Astrodynamics</td>
</tr>
<tr>
<td>Dr. Motee</td>
<td>LU/Mechanical</td>
<td>Committee Member</td>
<td>Expert in control theory, autonomous vehicles, &amp; optimization</td>
</tr>
<tr>
<td>Dr. DeLeo</td>
<td>LU/Physics</td>
<td>Committee Member</td>
<td>Expert in mass flow &amp; UV spectroscopy in binary-stars</td>
</tr>
</tbody>
</table>

Table 2: Ph.D. Committee

Both the department and the college require that the committee has at least four members, three of which have voting rights [1, 2]. Mr. Abraham’s committee is comprised of five members, three of whom have voting rights.

2.3 Department of Mechanical Engineering & Mechanics Requirements

According to [1] the departmental requirements for the doctoral degree in Mechanical Engineering focus on the student obtaining a minimum GPA in five core courses as well as a General Examination administered by the Ph.D. Committee.

GPA Minimum

According to [1] students are admitted to Ph.D. Candidacy only after obtaining a minimum average GPA of 3.35 in five pre-approved core courses. Mr. Abraham has completed six pre-approved courses as listed in Table 3.

<table>
<thead>
<tr>
<th>Course</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>Math Physics, Phy 428</td>
<td>4.00</td>
</tr>
<tr>
<td>Math Methods II, ME 453</td>
<td>3.30</td>
</tr>
<tr>
<td>Heat &amp; Mass Trans., ME 423</td>
<td>3.00</td>
</tr>
<tr>
<td>Fluid Mechanics, ME 430</td>
<td>3.70</td>
</tr>
<tr>
<td>Manufacturing, ME 402</td>
<td>4.00</td>
</tr>
<tr>
<td>Dynamics, MECH 425</td>
<td>4.00</td>
</tr>
<tr>
<td><strong>Average:</strong></td>
<td><strong>3.67</strong></td>
</tr>
</tbody>
</table>

Table 3: GPA of Five Core-Courses

As is obvious from Table 3 Mr. Abraham has exceeded the minimum 3.35 GPA requirement. Even if the lowest five core courses were averaged it would still exceed this requirement.
General Examination

On Friday, August 31, 2012 Mr. Abraham successfully passed his General Examination. He has unanimously received a passing vote from all the committee members: Dr. Blythe, Dr. Spencer, Dr. DeLeo, Dr. Motee, and Dr. Hart.

2.4 P.C. Rossin College of Engineering and Applied Science Requirements

Among the plethora of requirements found in [2], two have relevance to this document. The first is that the student must submit a Ph.D. Proposal (which is this document) as well as an application of candidacy form. The second is that the student must complete 72 credit hours before admittance to Ph.D. Candidacy is allowed. As of the Fall 2012 semester, Mr. Abraham is on track to finish 73 credit hours by the end of the semester; 58 of which is coursework. Once admitted into candidacy, Mr. Abraham will take one “Maintenance of Candidacy” credit per semester (fall and spring) until graduation.

3 State of the Art in Trajectory Design

Currently there are multiple tools available for mission planners to use when investigating low-thrust trajectories. NASA has organized many of these tools into a Low Thrust Trajectory Tool (LTTT) set led by Marshal Space Flight Center [24, 15]. The tools included in LTTT include: Copernicus, MALTO, Mystic, OTIS, and SNAP. Other software that is currently not part of the LTTT include CHEBYTOP and VARITOP/SEPTOP/NEWSEP. To date, Mr. Abraham has not acquired any of these software tools but is in active pursuit of MALTO and Copernicus.

3.1 Copernicus

The Copernicus software is a general-purpose trajectory design and optimization system originally developed by the University of Texas in 2001 and later developed by NASA’s Johnson Space Center and Goddard Space Flight Center. It is a highly comprehensive software that is capable of producing high-precision trajectories of single or multiple spacecraft in an N-body gravitational environment [18]. It can accommodate a variety of propulsion systems including finite and impulsive propulsive maneuvers. The software is capable of using a variety of (user selected) algorithms to optimize spacecraft trajectories and includes 3-D visualization of these trajectories.

3.2 MALTO

The Mission Analysis Low-Thrust Optimization (MALTO) tool was designed by NASA Jet Propulsion Laboratory for quick trade studies of low-thrust missions. The code is set up to handle only two-body dynamics (primarily heliocentric trajectories but geocentric is also allowed) but it does so with robust convergence. This is accomplished by dividing the trajectory into $n$ segments and assigning a small but finite $\Delta V$ to each segment, thus discretizing the otherwise continuous low-thrust [23]. An optimization algorithm is then run to “patch” the ends of each segment together (in both position and velocity) in order to get a continuous, low-thrust arc as the final product.

3.3 Mystic

Mystic was developed at NASA Jet Propulsion Laboratory by Dr. Greg Whiffen and others. It performs nonlinear optimization by using Static/Dynamic optimal control. Mystic is a high-fidelity tool that can incorporate n-body dynamics into low-thrust trajectories [28, 15]. Mystic specializes in interplanetary trajectories, especially those that use a gravity-assist maneuver. The software has the ability to automatically target and utilize gravity assists in a mission planning scenario [13]. This software has been validated using trajectories from the Dawn mission [13].
3.4 OTIS

The Optimal Trajectories by Implicit Simulation (OTIS) software was a joint product developed by NASA Glenn Research Center and The Boeing Company. Initially, the program focused on launch vehicle ascent trajectories, but has since been generalized to include interplanetary missions with low-thrust propulsion [22, 10, 13]. The program is capable of high-fidelity optimization of spacecraft trajectories using up to six degrees of freedom. The program is free to use for U.S. citizens but is subject to ITAR restrictions [15].

3.5 SNAP

The Spacecraft N-body Analysis Program (SNAP) is a high-fidelity trajectory optimization propagator [15]. It is capable of accounting for trajectory perturbations such as solar radiation pressure, atmospheric drag, NxN gravity harmonics, and low-thrust maneuvering including shadowing effects. SNAP uses a Runge-Kutta Fehlberg method of order 7-8 as its numerical integrator for trajectory propagation [13, 16].

3.6 CHEBYTOP

The Chebyshev Trajectory Optimization Program (CHEBYTOP) is a general-purpose, two-body, heliocentric, low-thrust trajectory optimization software created by NASA Glenn Research Center and Boeing [14, 15]. The program runs relatively quickly and was primarily designed for low-thrust, interplanetary trajectories. It is a collocation-based trajectory optimization code that assumes variable thrust will behave in a way similar to constant thrust trajectories for a short enough period of time [14].

3.7 VARITOP/SEPTOP/NEWSEP

The Variational calculus Trajectory Optimization Program (VARITOP) was developed by Carl Sauer at JPL and later updated by new additions to the code called SEPTOP and NEWSEP. VARITOP uses the calculus of variations, primer vector theory, and Pontrygin’s maximum principle to solve the optimization problem [13]. The Solar Electric Propulsion Trajectory Optimization Program (SEPTOP) and it’s update, NEWSEP, focus on low-thrust trajectories. They can simulate the influence of low-thrust propulsion using polynomial expansions and also take into account the decrease in thrust associated with solar array performance decreases as a function of time and distance from the sun [13].

3.8 LTool

NASA’s Jet Propulsion Laboratory has created a software tool called the Libration Point Mission Design Tool (LTool) that was primarily used for the Genesis Mission. LTool is specifically designed to utilize three-body dynamics found in Lagrange-point orbits. It can successfully model Halo, Lissajous, and Lyapunov orbits. Unfortunately, LTool was originally intended for short, finite burns (chemical propulsion) and is not optimized for low-thrust trajectories [11].

4 Optimization Algorithms and Force Models

This section describes much of the background knowledge needed to understand both the current and future work proposed in this document. This dissertation will rely heavily on two important topics found in Astrodynamics. One such topic is the Circular Restricted Three Body Problem (CR3BP). Detailed information on this topic can be found in the Appendix and serves as the foundation for much of the remainder of this proposal.
4.1 Coordinate System and Definitions

This is a derivation of the Circular Restricted 3-Body Problem (CR3BP). In this derivation, an attempt will be made to non-dimensionalize as many quantities as possible in order to simplify the system. First, set the distance between Primary 1 and Primary 2 to be equal to one “distance unit,” \( d_u \). Next, define the reduced mass as:

\[
\mu \equiv \frac{m_2}{m_1 + m_2} \tag{1}
\]

Furthermore, choose the convention \( m_1 \geq m_2 \gg m_3 \). It will be assumed that the mass of the third body is so insignificant in comparison to Primary 1 and Primary 2 that it does not affect the motion of either Primary (hence the “restriction” in the CR3BP). Due to this assumption, it can be stated that \( m_1 + m_2 + m_3 \cong m_1 + m_2 \). Furthermore, for the sake of simplicity, define the units of mass in such a way as to set the total mass of the system; \( m_1 + m_2 = 1 \). The Center Of Mass (C.O.M.) of the system (commonly known as the barycenter in the Astrodynamics community) can be calculated in the usual way:

\[
C.O.M.\text{ Relative to } m_1 = \frac{m_1}{m_1 + m_2}(0) + \frac{m_2}{m_1 + m_2}(1) = m_2 = \mu
\]

or

\[
C.O.M.\text{ Relative to } m_2 = 1 - C.O.M.\text{ Relative to } m_1 = 1 - \mu
\]

A coordinate system is chosen with the barycenter as its origin as can be seen in Figure 1. The positive \( x \)-axis is defined as the direction from the larger primary to the smaller primary. The positive \( y \)-axis is defined to be in the same direction as velocity vector of the smaller primary (which is perpendicular to the \( x \)-axis due to the circular nature of its orbit). Finally, the positive \( z \)-axis is defined in such a way as to complete the right-handed triad based on the \( x \) and \( y \) unit vectors. A non-inertial reference frame will be utilized; this frame shall rotate with an angular velocity that exactly matches that of the two primaries which are orbit about their common barycenter. Note that \( m_1 \) is located at \((-\mu, 0, 0)\) and \( m_2 \) at \((1 - \mu, 0, 0)\) in this frame. This co-rotating reference frame restricts the two primaries to remain in fixed
positions along the $x$-axis. A non-inertial reference frame may seem to be (at first glance) a needless complication, but this is not the case and shall prove to be very beniful in a later analysis.

4.2 System Dynamics

The kinetic energy and potential energy of the third body can be written as:

$$T = \frac{1}{2} m_3 v^2 = \frac{1}{2} m_3 \left[ (\dot{x} - \omega y)^2 + (\dot{y} + \omega x)^2 + \dot{z}^2 \right]$$

$$V = -\frac{G m_3 m_1}{r_{13}} - \frac{G m_3 m_2}{r_{23}}$$

$$E = T + V$$

$$E = m_3 \left( \frac{1}{2} \left[ (\dot{x} - \omega y)^2 + (\dot{y} + \omega x)^2 + \dot{z}^2 \right] - \frac{G m_3}{r_{13}} - \frac{G m_2}{r_{23}} \right)$$

with the distances $r_{13} = \sqrt{(x + \mu)^2 + y^2 + z^2}$ and $r_{23} = \sqrt{(x - [1 - \mu])^2 + y^2 + z^2}$. Note that the total energy of the system needs further simplification. From a thorough study of Kepler’s Laws and Newtonian Mechanics, it can be shown that the angular velocity $\omega$ of two, circular, co-rotating masses is given by:

$$\omega = \sqrt{\frac{G (m_1 + m_2)}{r_{12}}}$$

This can be further simplified by utilizing the dimensionless units and quantities defined above:

$$\omega = \sqrt{\frac{G(1)}{1}} = \sqrt{G}$$

Notice that the dimensionless definitions helped to significantly reduce the angular velocity equation. Further reduction of this equation is possible by cleverly defining the dimensionless time units $\text{[tu]}$ of the system to be:

$$1 \text{[tu]} = \frac{T}{2\pi}$$

with $T$ being the period of the orbits of the primaries. This substitution will define

$$\sqrt{G} = \omega = \frac{2\pi}{2\pi} \left[ \frac{\text{rad}}{\text{[tu]}} \right] = 1$$

This will simplify the total energy per unit mass as:

$$\frac{E}{m_3} = \varepsilon = \frac{1}{2} \left[ (\dot{x} - y)^2 + (\dot{y} + x)^2 + \dot{z}^2 \right] - \frac{1 - \mu}{r_{13}} - \frac{\mu}{r_{23}}$$

We can also define the Lagrangian of the system as:

$$T = \frac{1}{2} m_3 \left[ (\dot{x} - y)^2 + (\dot{y} + x)^2 + \dot{z}^2 \right]$$
\[ V = -m_3 \left\{ \frac{1 - \mu}{\sqrt{(x + \mu)^2 + y^2 + z^2}} + \frac{\mu}{\sqrt{(x - [1 - \mu])^2 + y^2 + z^2}} \right\} \]

\[ L = T - V \]

\[ L = m_3 \left\{ \frac{1}{2} \left[ (\dot{x} - y)^2 + (\dot{y} + x)^2 + \dot{z}^2 \right] + \frac{1 - \mu}{\sqrt{(x + \mu)^2 + y^2 + z^2}} + \frac{\mu}{\sqrt{(x - [1 - \mu])^2 + y^2 + z^2}} \right\} \]

Using the Euler-Lagrange equations one can solve for the equations of motion:

\[ \frac{\partial L}{\partial x} - \frac{d}{dt} \frac{\partial L}{\partial \dot{x}} = 0 \]

\[ \dot{y} + x - \frac{\partial V}{\partial x} - \frac{d}{dt} [\dot{x} - y] = 0 \]

\[ \dot{x} - 2\dot{y} = -\frac{\partial V}{\partial x} + x \]

(2)

\[ \frac{\partial L}{\partial y} - \frac{d}{dt} \frac{\partial L}{\partial \dot{y}} = 0 \]

\[ - (\dot{x} - y) - \frac{\partial V}{\partial y} - \frac{d}{dt} [\dot{y} + x] = 0 \]

\[ \dot{y} + 2\dot{x} = -\frac{\partial V}{\partial y} + y \]

(3)

\[ \frac{\partial L}{\partial z} - \frac{d}{dt} \frac{\partial L}{\partial \dot{z}} = 0 \]

\[ - \frac{\partial V}{\partial z} - \frac{d}{dt} [\dot{z}] = 0 \]

\[ \ddot{z} = -\frac{\partial V}{\partial z} \]

(4)

As one further simplification, the potential function present in Equation 2 and Equation 3 can be re-defined. Let:

\[ U = V + f(x,y) \]

\[ -\frac{\partial U}{\partial x} \implies -\frac{\partial V}{\partial x} - \frac{\partial f}{\partial x} = -\frac{\partial V}{\partial x} + x \implies \frac{\partial f}{\partial x} = -x \implies f = -\frac{1}{2}x^2 + g(y) \]

\[ -\frac{\partial U}{\partial y} \implies -\frac{\partial V}{\partial y} - \frac{\partial f}{\partial y} = -\frac{\partial V}{\partial y} + y \implies \frac{\partial f}{\partial y} = -y \implies f = -\frac{1}{2}y^2 + h(x) \]

\[ \therefore f = -\frac{1}{2} (x^2 + y^2) \implies U = V - \frac{1}{2} (x^2 + y^2) \]
with $U$ known as the “pseudopotential.” Now Equation (2), (3), and (4) simplify to:

\[
\begin{align*}
\ddot{x} - 2\dot{y} &= -\frac{\partial U}{\partial x} \\
\ddot{y} + 2\dot{x} &= -\frac{\partial U}{\partial y} \\
\ddot{z} &= -\frac{\partial U}{\partial z}
\end{align*}
\]

which are the equations of motion of the system. The equations of motion have five equilibrium points, $L_1$-$L_5$, associated with them and shown in Figure 2 for a low value of $\mu$. Refer to the appendix for a detailed discussion of these equilibrium points.

![Figure 2: Five Lagrange Points for a Low-$\mu$ System](image)

### 4.3 State Transition Matrix

When undertaking numerical calculations it is often necessary to use gradient-based differential correction methods. Many of these methods rely on something known as the State Transition Matrix (STM). The STM relates the sensitivity of the final state $X_f(t_0, t)$ (obtained via integration of the EOM) to the initial state $X_0$. In other words the STM $\equiv \Phi_{(t,t_0)} = \frac{\partial X_f}{\partial X_0}$, or
\[ \Phi(t,t_0) = \begin{bmatrix} \frac{\partial x}{\partial t} & \frac{\partial x}{\partial t_0} & \frac{\partial x}{\partial y} & \frac{\partial x}{\partial y_0} & \frac{\partial x}{\partial z} & \frac{\partial x}{\partial z_0} & \frac{\partial x}{\partial ɬ} & \frac{\partial x}{\partial ɬ_0} & \frac{\partial x}{\partial ɬ} & \frac{\partial x}{\partial ɬ_0} \\ \frac{\partial x}{\partial t_0} & \frac{\partial x}{\partial y} & \frac{\partial x}{\partial y_0} & \frac{\partial x}{\partial z} & \frac{\partial x}{\partial z_0} & \frac{\partial x}{\partial ɬ} & \frac{\partial x}{\partial ɬ_0} & \frac{\partial x}{\partial ɬ} & \frac{\partial x}{\partial ɬ_0} & \frac{\partial x}{\partial ɬ} & \frac{\partial x}{\partial ɬ_0} \\ \frac{\partial x}{\partial z} & \frac{\partial x}{\partial z_0} & \frac{\partial x}{\partial ɬ} & \frac{\partial x}{\partial ɬ_0} & \frac{\partial x}{\partial ɬ} & \frac{\partial x}{\partial ɬ_0} & \frac{\partial x}{\partial ɬ} & \frac{\partial x}{\partial ɬ_0} & \frac{\partial x}{\partial ɬ} & \frac{\partial x}{\partial ɬ_0} & \frac{\partial x}{\partial ɬ} \\ \frac{\partial x}{\partial ɬ} & \frac{\partial x}{\partial ɬ_0} & \frac{\partial x}{\partial ɬ} & \frac{\partial x}{\partial ɬ_0} & \frac{\partial x}{\partial ɬ} & \frac{\partial x}{\partial ɬ_0} & \frac{\partial x}{\partial ɬ} & \frac{\partial x}{\partial ɬ_0} & \frac{\partial x}{\partial ɬ} & \frac{\partial x}{\partial ɬ_0} & \frac{\partial x}{\partial ɬ} \\ \frac{\partial x}{\partial ɬ} & \frac{\partial x}{\partial ɬ_0} & \frac{\partial x}{\partial ɬ} & \frac{\partial x}{\partial ɬ_0} & \frac{\partial x}{\partial ɬ} & \frac{\partial x}{\partial ɬ_0} & \frac{\partial x}{\partial ɬ} & \frac{\partial x}{\partial ɬ_0} & \frac{\partial x}{\partial ɬ} & \frac{\partial x}{\partial ɬ_0} & \frac{\partial x}{\partial ɬ} \end{bmatrix} \] \tag{6}

This can be demonstrated in the following way. Suppose two trajectories exist: trajectory \( X \), and its variation, trajectory \( Y \) (i.e. \( Y \) is in the neighborhood of \( X \)). According to [27], one can express the derivatives and initial conditions as:

\[
\begin{align*}
X(t_0) &= X_0 \\
\dot{X} &= \vec{f}(X) \\
Y(t_0) &= Y_0 \\
\dot{Y} &= \vec{f}(Y)
\end{align*}
\tag{7}
\]

where \( \vec{f} \) has the same definition as that found in Section 11 of the appendix. The difference in these two trajectories is small and can be expressed as:

\[
Y = X + \delta x
\tag{8}
\]

By taking the time derivative of Equation 8 and substituting from Equation 7 one can define \( \dot{Y} \) as

\[
\dot{Y} = \vec{f}(X + \delta x)
\]

and when expanded using a Taylor series centered about \( X \), using Equation 8

\[
\dot{Y} = \vec{f}(X) + \frac{\partial \vec{f}(X)}{\partial X} \delta x + \frac{\partial^2 \vec{f}(X)}{2!} \delta x^2 + ... \tag{9}
\]

Note that \( \dot{Y} = \dot{X} + \delta \dot{x} \) via a differentiation of Equation 8 and when substituted into Equation 9 yields

\[
\delta \dot{x} = \frac{\partial \vec{f}(X)}{\partial X} \delta x + u = A(t) \delta x + u \tag{10}
\]

where [27] uses \( u \) to represent all the higher order terms of the expansion. If \( u \) is ignored one can say that the system has been “linearized” because the non-linear terms have been disregarded. This approximation is valid in regions where the dynamics are only weakly non-linear such as the vicinity of a Lagrange point.

Assume that a solution to (differential) Equation 10 has the form

\[
\delta x = \Phi(t,t_0) \delta x_0 \tag{11}
\]

If Equation 11 is inserted into Equation 10 the result is

\[
\Phi(t,t_0) = A(t) \Phi(t,t_0) \tag{12}
\]

which can be numerically integrated from an initial condition to determine the STM at any moment in time. The initial condition is found by plugging in \( t = t_0 \) into Equation 11. When doing so, one will find that \( \Phi(t_0,t_0) = I \).
4.4 Generalized Free Variable & Constraint Vectors

When using a differential corrections algorithm to solve a two-point boundary value problem (TPBVP) one can use free variable and constraint vectors based on Newton's method \[9, 20\]. The free variable vector

\[
X = \begin{bmatrix}
X_1 \\
\vdots \\
X_n
\end{bmatrix}
\]

consists of \(n\) elements that are allowed to freely and independently change in value during the course of a differential corrections procedure. Typically, these elements consist of the state vector, integration times, \(T_i\), and slack variables. The system is also subject to \(m\) constraint equations. These constraint equations form the Constraint Vector

\[
F(X) = \begin{bmatrix}
F_1(X) \\
\vdots \\
F_m(X)
\end{bmatrix} = 0
\]

and must be written in such a way that they satisfy the equation \(F(X) = 0\).

The goal is to find a solution, \(X^*\), which satisfies the equation \(F(X^*) = 0\) given an acceptable error tolerance, \(\varepsilon\). In order to accomplish this task, one may begin with an initial free variable vector \(X^0\). One can express \(F(X)\) by using a Taylor Series centered about \(X^0\) and dropping all higher order terms

\[
F(X) = F(X^0) + DF(X^0)(X - X^0)
\]

with the \(m \times n\) Jacobian Matrix \(DF(X^0)\) expressed as

\[
DF(X^0) = \frac{\partial F(X^0)}{\partial X^0} = \begin{bmatrix}
\frac{\partial F_1}{\partial X_1} & \cdots & \frac{\partial F_1}{\partial X_n} \\
\vdots & \ddots & \vdots \\
\frac{\partial F_m}{\partial X_1} & \cdots & \frac{\partial F_m}{\partial X_n}
\end{bmatrix}_{X=X^0}
\]

Next an iterative process will be introduced into Equation 13. Note that the substitution \(F(X) = 0\) can be used to reduce Equation 13 to

\[
0 = F(X^j) + DF(X^j)(X^{j+1} - X^j)
\]

where \(X^j\) represents the current iteration and \(X^{j+1}\) represents the next iteration. If the algorithm is convergent, then \(\|F(X^j)\| > \|F(X^{j+1})\|\) and will stop iterating once \(\|F(X^j)\| < \varepsilon\).

There are two ways to solve for \(X^*\) depending on the situation. If \(n = m\) then \(DF\) is square and invertible. In this situation a multivariate version of Newton’s Method is appropriate:

\[
X^{j+1} = X^j - DF^{-1}(X^j)F(X^j)
\]

If, however, \(n > m\), there exists an infinite number of solutions and \(DF\) cannot be inverted. In this circumstance the minimum-norm solution is used:

\[
X^{j+1} = X^j - DF^T(X^j)\left[DF(X^j)DF^T(X^j)\right]^{-1}F(X^j)
\]

This solution minimizes the difference between the current iteration and the previous one. This is desirable because it typically finds the solution, \(X^*\), that is closest to the initial guess, \(X^0\), given the fact that there are an infinite number of solutions.

In summary, to solve a general optimization problem using free variable and constraint vectors one must:
1. Define $X$ and $F(X) = 0$

2. Calculate the Jacobian Matrix, $DF(X)$, and the associated partial derivatives therein.

3. Iteratively solve for $X^*$ using either Equation 15 or Equation 16 depending on the relationship between $n$ and $m$.

4.5 Variable-Time, Single Shooting

![Figure 3: Variable-Time, Single Shooting](image)

One application of free variables and constraint vectors is that of Variable-Time, Single Shooting [20]. In this application the position of the initial state of a trajectory is fixed but the initial velocity and Time of Flight, $T$, are free. Thus the free variable vector becomes

$$X = \begin{bmatrix} \dot{x}_0 \\ \dot{y}_0 \\ \dot{z}_0 \\ T \end{bmatrix}$$

(17)

The constraint vector will be written as

$$F(X) = \begin{bmatrix} x_T - x_d \\ y_T - y_d \\ z_T - z_d \end{bmatrix}$$

(18)

with $X_d$ representing the desired state at the end of time $T$. The Jacobian Matrix can be represented as

$$DF(X) = \frac{\partial F(X)}{\partial X} = \begin{bmatrix} \frac{\partial x_T}{\partial x_0} & \frac{\partial x_T}{\partial y_0} & \frac{\partial x_T}{\partial z_0} & \frac{\partial x_T}{\partial T} & \frac{\partial \dot{x}_T}{\partial x_0} & \frac{\partial \dot{x}_T}{\partial y_0} & \frac{\partial \dot{x}_T}{\partial z_0} & \frac{\partial \dot{x}_T}{\partial T} \\ \frac{\partial y_T}{\partial x_0} & \frac{\partial y_T}{\partial y_0} & \frac{\partial y_T}{\partial z_0} & \frac{\partial y_T}{\partial T} & \frac{\partial \dot{y}_T}{\partial x_0} & \frac{\partial \dot{y}_T}{\partial y_0} & \frac{\partial \dot{y}_T}{\partial z_0} & \frac{\partial \dot{y}_T}{\partial T} \\ \frac{\partial z_T}{\partial x_0} & \frac{\partial z_T}{\partial y_0} & \frac{\partial z_T}{\partial z_0} & \frac{\partial z_T}{\partial T} & \frac{\partial \dot{z}_T}{\partial x_0} & \frac{\partial \dot{z}_T}{\partial y_0} & \frac{\partial \dot{z}_T}{\partial z_0} & \frac{\partial \dot{z}_T}{\partial T} \end{bmatrix} = \begin{bmatrix} \Phi_{1.4} & \Phi_{1.5} & \Phi_{1.6} & \dot{x}_T \\ \Phi_{2.4} & \Phi_{2.5} & \Phi_{2.6} & \dot{y}_T \\ \Phi_{3.4} & \Phi_{3.5} & \Phi_{3.6} & \dot{z}_T \end{bmatrix}$$

(19)
in accordance with Equation 6. Since the Jacobian is not a square matrix, the minimum-norm solution may be found using Equation 16.

4.6 Variable-Time, Multiple Shooting

![Diagram of Variable-Time, Multiple Shooting](image)

**Figure 4: Variable-Time, Multiple Shooting**

Next the concept of Variable-Time, Multiple Shooting will be explored. This is a highly useful technique to use in finding sensitive trajectories because a number of “way-points” can be used between the initial and final states [20]. Figure 4 shows a trajectory that is beginning to form using the Variable-Time, Multiple Shooting method. This trajectory is comprised of \( n \) segments with each segment being numerically propagated from its initial state, \( X_n \), to its final state, \( X_{n+1}, T_n \). A converged trajectory needs to be fully continuous along its entire path. It is therefore necessary for the final state of a segment to match the initial state of its succeeding segment. To begin, a free variable vector is defined as

\[
X = \begin{bmatrix}
X_1 \\
X_2 \\
\vdots \\
X_n \\
T_1 \\
\vdots \\
T_{n-1}
\end{bmatrix}
\]  

(20)

with \( 7n - 1 \) components. A constraint vector is defined as

\[
F(X) = \begin{bmatrix}
X_{2,T_1} - X_2 \\
\vdots \\
X_{n,T_{n-1}} - X_n
\end{bmatrix}
\]  

(21)
which dictates the need for each new segment of the trajectory to start/stop with an identical state as its preceding/proceeding segment. The constraint vector has \(6(n-1)\) components. The Jacobian can be expressed as

\[
\begin{pmatrix}
\Phi_{(r_2,r_1)} & -I_{6\times6} & 0_{6\times6} & \cdots & \cdots & 0_{6\times6} & \hat{X}_{2,T_1(6\times1)} & 0_{6\times1} & \cdots & 0_{6\times1} \\
0_{6\times6} & \Phi_{(r_3,r_2)} & -I_{6\times6} & 0_{6\times6} & \cdots & \cdots & \cdots & 0_{6\times1} & \hat{X}_{3,T_2(6\times1)} & \cdots & \cdots \\
\vdots & 0_{6\times6} & \Phi_{(r_4,r_3)} & -I_{6\times6} & 0_{6\times6} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
0_{6\times6} & 0_{6\times6} & \ddots & 0_{6\times6} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
\vdots & \vdots & \ddots & 0_{6\times6} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
\vdots & \vdots & \ddots & \Phi_{(r_{n-1},r_{n-2})} & -I_{6\times6} & 0_{6\times6} & \cdots & 0_{6\times1} & 0_{6\times1} & \cdots & 0_{6\times1} \\
0_{6\times6} & 0_{6\times6} & \cdots & \Phi_{(r_{n},r_{n-1})} & -I_{6\times6} & 0_{6\times6} & \cdots & 0_{6\times1} & 0_{6\times1} & \cdots & 0_{6\times1} \\
0_{6\times6} & 0_{6\times6} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
0_{6\times6} & 0_{6\times6} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
\end{pmatrix}
\]

(22)

which is a \(6(n-1) \times 7n-1\) matrix. This method is generalized to account for \(n\) way-points that flow together using a continuous trajectory with a total time of \(T = \sum_{i=1}^{n} T_i\).

### 4.7 Single Parameter Continuation

The underlying concept behind Single-Parameter Continuation is relatively straightforward. Assuming a (desired) baseline trajectory can be found using a standard optimization method, such as those outlined above, it should also be possible to use the same techniques to find related solutions. This can be accomplished by using the same initial state as was used with the baseline solution but with a small change in one of the parameters. This new initial state will then be run through the given optimization method until a new solution is reached. This new solution will differ from the baseline trajectory because of the small change in one of the parameters of the initial state. This process can be repeated numerous times to build up a “family” of trajectories that are all based on the “baseline” trajectory and a continuation of a parameter of interest. Traditionally, parameters that are changed can involve position, velocity, energy, period, and so on. Normally, only one parameter is used in this continuation scheme.

### 4.8 Pseudo-Arclength Continuation

In contrast to Single Parameter Continuation, Pseudo-Arclength Continuation can vary multiple parameters during one iteration. The way in which these parameters are altered cause the solution to move in a direction that is tangent to the baseline solution by a (user defined) amount \(\Delta s\). In this way the continuation scheme moves away from the current family of trajectories in search for new families of trajectories [20].

Let \(X_{i-1}^*\) be a (converged) baseline trajectory that satisfies the equation \(F(X_{i-1}^*) = 0\). The goal is to find the next member of the continuation scheme, \(X_i\), which often defines a new family of trajectories. The null vector of the Jacobian Matrix, \(DF(X_{i-1}^*)\), is used to generate an orthonormal null vector, \(\Delta X_{i-1}\), that is tangent to the family at \(X_{i-1}^*\). To ensure that the new member, \(X_i\), is found by moving an amount \(\Delta s\) in the tangent direction a pseudo-arclength constraint is appended to the existing constraint vector, \(F(X_i)\). The new (augmented) constraint vector is written as

\[
G(X_i) = \begin{bmatrix}
F(X_i) \\
(X_i - X_{i-1}^*)^T \Delta X_{i-1} - \Delta s
\end{bmatrix} = 0
\]

(23)
and the augmented Jacobian Matrix is written as

\[ DG_{(X_i)} = \frac{\partial G_{(X_i)}}{\partial X_i} = \begin{bmatrix} DF_{(X_i)} \\ (\Delta X_{i-1})^T \end{bmatrix} \] (24)

These equations can be solved using either Newton’s Method or Equations [16]. This process can be repeated numerous times to generate a large number of new families of trajectories (as desired).

4.9 Full Ephemeris

While the equations of motion in the CR3BP are a useful starting point for accurate trajectory propagation they are not accurate enough for high-fidelity modeling. Perturbations from other gravitational bodies (i.e. Sun, Venus, Jupiter, etc.) can cause significant changes in the trajectory of a satellite in High Earth Orbit (HEO). These perturbations become increasingly important in the vicinity of Lagrange points since much of the three-body effects cancel and the resultant force is astonishingly small.

In order to accurately account for the gravitational perturbations of extra bodies the exact position of each body with respect to either some reference point or the spacecraft itself must be known at a given epoch. Once this information is known, the following equation can be used to determine the acceleration on a spacecraft from \( n \) gravitational bodies [27]

\[ \ddot{r}_{1\text{sat}} = \frac{-G (m_1 + m_{\text{sat}})}{r_{1\text{sat}}^3} r_{1\text{sat}} + G \sum_{j=3}^{n} m_j \left( \frac{r_{\text{sat}j}}{r_{\text{sat}j}^3} - \frac{r_{1j}}{r_{1j}^3} \right) \] (25)

4.10 SPICE/Mice

If a full ephemeris force model is to be employed, the exact positions of all major gravitational objects in the Solar System must be known at a given epoch. While it is possible to use Keplerian models to derive algebraic approximations of the positions of these bodies, it is much more accurate to glean this information from direct astronomical observation. This is where the capabilities of the SPICE kernel produced by NASA’s Navigation and Ancillary Information Facility (NAIF) becomes paramount [17]. The primary goal of SPICE (Spacecraft, Planet, Instrument, C-Matrix “pointing”, and Events kernel) is to define, develop, and utilize software standards/protocols that can store data gathered from spacecraft missions from any agency, nation, or organization in a uniform way.

One of the key functions of SPICE is its ability to quickly retrieve ephemeris data (i.e. position and velocity relative to a coordinate system at a given moment in time) from any time period between the 1970’s and the 2050’s. All data is based on astronomical observation, spacecraft reconnaissance, or advanced mathematical modeling where appropriate. This data represents some of the most accurate ephemeris data available to the public and is used as the standard in many software applications [3]. This SPICE kernel is supported by four major programming languages: C, FORTRAN, IDL, and MATLAB. The MATLAB version, called “Mice” is the primary language used for this dissertation work.

5 Current Work

This section will describe all of the work that has been done by Mr. Abraham over the past few months. Programs that are authored by Mr. Abraham shall be discussed and data generated by these programs will be displayed.

5.1 Variable Time, Single Shooting Program

The goal of the Variable Time, Single Shooting program is to compute trajectories that are closed and periodic about the Earth-Moon \( L_2 \) point. This program will take advantage of the symmetry in the dynamics found across the xz-plane
and restrict the motion of the spacecraft to the xy-plane (for simplicity). The xz-plane symmetry will ensure that any trajectory which begins at the x-axis and then crosses the x-axis with a velocity vector tangent to the x-axis (and still within the xy-plane), must produce a new trajectory that is a reflection of the original about the x-axis [6]. This is diagrammed in Figure 5.

![Figure 5: Reflection of Converged Trajectory Across X-Axis](image)

This program uses a Free Variable Vector of the form

$$X = \begin{bmatrix} x_0 \\ y_0 \\ \frac{1}{2}T \\ \beta \end{bmatrix}$$  \hspace{1cm} (26)

Note that the first two entries are the initial x-position and initial y-velocity (assumed to be positive). All other components of the state vector are assumed to be zero. This implies that the initial position is located somewhere on the x-axis. The third component of the Free Variable Vector is the half-period of the orbit. This indicates that the intention of the shooting algorithm is to find a half-orbit rather than a full orbit. Finally, if the initial velocity is positive, then the velocity at the end of a half-period must be negative (or else there would not be a closed orbit). This implies a constraint condition of $\dot{y}_{\frac{1}{2}T} < 0$. Unfortunately, this constraint condition cannot be written into the constraint vector in its present form. Instead the condition must be modified to

$$\dot{y}_{\frac{1}{2}T} + \beta^2 = 0$$  \hspace{1cm} (27)

with $\beta$ being real and known as the “slack variable.” Since $\beta$ is real the condition above can be expressed in the Constraint Vector as

$$F(X) = \begin{bmatrix} \dot{y}_{\frac{1}{2}T} \\ \dot{x}_{\frac{1}{2}T} \\ \dot{y}_{\frac{1}{2}T} + \beta^2 \end{bmatrix} = 0$$  \hspace{1cm} (28)
and the Jacobian Matrix as

\[
DF(X) = \frac{\partial F(X)}{\partial X} = \begin{bmatrix}
\frac{\partial y_1}{\partial x_0} & \frac{\partial y_2}{\partial x_0} & \frac{\partial y_3}{\partial x_0} & \frac{\partial y_4}{\partial x_0} & \frac{\partial y_5}{\partial x_0} \\
\frac{\partial y_1}{\partial y_1} & \frac{\partial y_2}{\partial y_1} & \frac{\partial y_3}{\partial y_1} & \frac{\partial y_4}{\partial y_1} & \frac{\partial y_5}{\partial y_1} \\
\frac{\partial y_1}{\partial \dot{y}_1} & \frac{\partial y_2}{\partial \dot{y}_1} & \frac{\partial y_3}{\partial \dot{y}_1} & \frac{\partial y_4}{\partial \dot{y}_1} & \frac{\partial y_5}{\partial \dot{y}_1} \\
\frac{\partial y_1}{\partial \dot{y}_2} & \frac{\partial y_2}{\partial \dot{y}_2} & \frac{\partial y_3}{\partial \dot{y}_2} & \frac{\partial y_4}{\partial \dot{y}_2} & \frac{\partial y_5}{\partial \dot{y}_2} \\
\frac{\partial y_1}{\partial \beta} & \frac{\partial y_2}{\partial \beta} & \frac{\partial y_3}{\partial \beta} & \frac{\partial y_4}{\partial \beta} & \frac{\partial y_5}{\partial \beta}
\end{bmatrix}
\]

(29)

where \( \Phi_{i,j} \) represents the \( i^{th} \) row and \( j^{th} \) column of the STM, \( \Phi_{(\frac{1}{2}T,0)} \). Note that the STM is numerically propagated from time 0 to time \( \frac{1}{2}T \) in order to populate the first six elements of the Jacobian Matrix. Three of the remaining elements are populated by evaluating the equations of motion at the final state of the trajectory.

As an example, the program was seeded with an initial state taken from tabulated values in [26]. The initial state was

\[
X_{0\text{guess}} = \begin{bmatrix}
0.78 & 0.00 & 0.00 & 0.00 & 0.443 & 0.000
\end{bmatrix}^T
\]

with an initial \( T_{\text{guess}} = 3.9 \). All values are in the non-dimensional units of the CR3BP. By the author’s experience, initial guess values of \( \beta = 0.7 \). The Variable-Time, Single Shooting algorithm ran for just over one second and used 10 iterations to produce a closed, periodic orbit about the Earth-Moon \( \text{L}_1 \) point with a tolerance of \( 10^{-12} \). The orbit can be seen in Figure 6.

Figure 6: Closed Earth-Moon \( \text{L}_1 \) Orbit Generated by Variable-Time, Single Shooting Method

The algorithm found the desired initial state to be

\[
X_{0\text{shot}} = \begin{bmatrix}
0.777910486548393 & 0.00 & 0.00 & 0.0 & 0.455080899040143 & 0.0
\end{bmatrix}
\]

with an orbital period of \( T = 4.063544575624369 \). Note the classic “kidney-bean” shape with curvature that is especially evident in the vicinity near the Moon (shown to the right of the orbit). Also note the 15 significant figures associated with each value reported (if the value is exactly zero, only two digits are reported for sake of simplicity). These 15 significant figures correspond to the machine precision of MATLAB’s 64 bit, double-precision, floating-point arithmetic. All values reported in this document should be assumed to be double-precision. This degree of precision is needed because the equations of motion are numerically integrated. As the integration time increases, so does the integration error, which is primarily due to limitations in machine precision.
5.2 Pseudo-Arclength Continuation & Single Shooting

The orbit found in the previous section by method of Single Shooting will now be used as a “seed trajectory” for a Pseudo-Arclength Continuation program. In this program $X$ was unchanged and

$$G(X) = \begin{bmatrix} F(X_i) \\ (X_i - X^*_i - 1)^T \Delta X^*_i - \Delta s \end{bmatrix} = \begin{bmatrix} y_{\frac{1}{2}}T \\ \dot{x}_{\frac{1}{2}}T \\ \dot{y}_{\frac{1}{2}}T + \beta^2 \\ (X_i - X^*_i - 1)^T \Delta X^*_i - \Delta s \end{bmatrix} = 0$$

with

$$DG(X_i) = \frac{\partial G(X_i)}{\partial X_i} = \begin{bmatrix} DF(X_i) \\ (\Delta X^*_i - 1)^T \end{bmatrix}$$

The baseline trajectory, $X^*_1$, is taken from the solution given by the Single Shooting algorithm from the previous section. Note that the orthonormal null vector $\Delta X^*_1$ was also calculated from the previous trajectory by using the MATLAB code $\text{deltaXstar} = \text{null}(DF)$. All subsequent null vectors are calculated using the same command. Based on trial-and-error experience, the value of $\Delta s = 0.012$ and was constant throughout the continuation process.

The code generated 100 orbits and ran for a total of 23 seconds. Every 10th orbit is plotted in Figure 7.

![Figure 7: Orbits Automatically Found Using Pseudo-Arclength Continuation (Outside Orbit = Baseline)](image)

5.3 Variable Time, Multiple Shooting Program

In this program Mr. Abraham uses the Multiple Shooting technique to find a trajectory that converges into periodic motion about the Earth-Moon $L_2$ point. The algorithm was tested using a known periodic solution which can be found in Table 4.
Table 4: 10 Known Patch Points

Note that the program is capable of accepting an arbitrary number of points, \( n \), with \( n \geq 3 \) but in this case \( n = 10 \). The solution in Table ?? was truncated to one digit after the decimal place for all values and then used as the baseline “patch points” for the Multiple Shooting algorithm. If the trajectory successfully converged it would prove that the Multiple Shooting algorithm was indeed a highly robust algorithm since the baseline trajectory was so badly damaged by data loss.

The Variable Time, Multiple Shooting algorithm followed most of the methodology outlined in the previous section with a few minor exceptions that allowed for a periodic (closed) trajectory instead of an open one. The free-variable vector was defined in the usual way as a \( 7n - 1 \) vector

\[
X = \begin{bmatrix}
X_1 \\
X_2 \\
\vdots \\
X_n \\
T_1 \\
\vdots \\
T_{n-1}
\end{bmatrix}
\]  

(32)

but the constraint vector was slightly different

\[
F(X) = \begin{bmatrix}
X_2 - X_1 \\
\vdots \\
X_{n-1} - X_{n-2} \\
X_n - T_{n-1} - X_1
\end{bmatrix}
\]  

(33)

but still a \( 6(n-1) \) vector. Note that the final patch-point is designed to match the initial patch-point as shown by the last line in Equation [35]. Also note that, on occasion, the program will not fully converge using this definition of the constraint vector. If this is the case then remove the constraint on the y-velocity component of the final patch point. Because of numerical integration error, it may become difficult for the solver to exactly match the initial patch point with the final patch point. The introduction of a “slack” variable, which allows a small amount of leeway in the y-velocity component, will typically resolve this issue. Finally, the
Jacobian matrix will change slightly to $DF(X) =$

\[
\begin{bmatrix}
\Phi(t_2, t_1) & -I_{6 \times 6} & 0_{6 \times 6} & \cdots & 0_{6 \times 6} & \dot{X}_2, T_1 (6 \times 1) & 0_{6 \times 1} & \cdots & 0_{6 \times 1} \\
0_{6 \times 6} & \Phi(t_3, t_2) & -I_{6 \times 6} & 0_{6 \times 6} & \cdots & \cdots & 0_{6 \times 1} & \cdots & 0_{6 \times 1} \\
\vdots & 0_{6 \times 6} & \Phi(t_4, t_3) & -I_{6 \times 6} & 0_{6 \times 6} & \cdots & \cdots & 0_{6 \times 1} & \cdots \\
\vdots & \vdots & \vdots & 0_{6 \times 6} & \cdots & \cdots & 0_{6 \times 1} & \cdots & 0_{6 \times 1} \\
I_{6 \times 6} & 0_{6 \times 6} & \cdots & \cdots & \cdots & 0_{6 \times 6} & \cdots & 0_{6 \times 1} & \cdots \\
0_{6 \times 6} & 0_{6 \times 6} & \cdots & \cdots & 0_{6 \times 6} & \cdots & 0_{6 \times 1} & \cdots & 0_{6 \times 1} \\
& & & & & & & & \\
\end{bmatrix}
\]

which is a $6(n - 1) \times 7n - 1$ matrix. Note the change in the matrix at the bottom left corner of Equation (34), from a null matrix to an identity matrix. This reflects the insertion of the closed-orbit boundary condition.

The results of this program can be seen in Figure 8. The thick lines indicate trajectory segments that were propagated using the 10 baseline patch points defined earlier. Obviously they are all highly undesirable and discontinuous trajectories. This program ran for 2.2 seconds with a total of 8 iterations before converging all segments/patch points to within $10^{-12}$ of each other. The converged patch points can be seen as color-coded stars with their corresponding trajectory segments of the same color. Notice that each trajectory segment matches the one before and after. In this way it has been demonstrated that a continuous trajectory can be formed from a highly fragmented trajectory using Variable-Time, Multiple Shooting techniques.

Figure 8: Data From Variable Time, Multiple Shooting Program
5.4 Example of Three-Dimensional Orbits

In general, a quasi-periodic trajectory can usually be found which maintains a spacecraft within a fixed volume of space, for a long period of time, in the vicinity of a Lagrange point. These trajectories are fully three-dimensional and may or may not consist of one or more closed orbits (typically they are nearly closed but not fully closed). These trajectories are called “lissajous” orbits. A special case of lissajous orbits exist when the orbits are fully closed and periodic. Such trajectories are known as “halo” orbits; named after the shape they often trace out when viewed by an Earth-bound observer looking directly at the Moon, and observing the shape of the spacecraft’s trajectory [7]. Finally, if the orbit is entirely two-dimensional and lies completely within the $xy$-plane of the CR3BP it is known as a “Lyapunov” orbit. All of the orbits displayed above are two-dimensional, Lyapunov orbits.

There, however, is no reason why a multiple shooting technique can’t be used to find halo orbits. Figure 9 shows the results of a multiple shooting program calculating a three-dimensional halo orbit. This orbit was found using the same program that was used in the previous section; only a periodic $z$-component (with small amplitude) was added to the initial conditions. The result was a fully converged halo orbit that has been plotted in three-dimensions (without the initial conditions displayed for the sake of simplicity). All ten segments begin at a colored star and propagate forward to the next trajectory segment (beginning with a different colored star). Note that the sphere just to the right of the orbit is the Moon; plotted to the correct scale.

![3-D plot](image)

**Figure 9: A Three-Dimensional Lagrange Point Orbit**

5.5 Full Ephemeris EOM Program with SPICE

The final program that was written attempted to utilize SPICE commands to accurately propagate trajectories under a full-ephemeris force model using Equation 25. Of course, it is not enough to simply make a trajectory propagator; it must be validated as well. Validation was accomplished by comparing two trajectories propagated using Abraham’s Ephemeris Model with identical trajectories propagated using Satellite Tool Kit (STK) software. STK software is a commercial product that has been on the market for many years and is well established as mature, tested, and validated [3]. Because of its established history STK is ideal benchmark to compare other software against.
The first trajectory is based on an arbitrary initial state of

\[
X_0 = \begin{bmatrix}
350000 \text{ [Km]} \\
0 \text{ [Km]} \\
0 \text{ [Km]} \\
0 \text{ [Km/s]} \\
0.5 \text{ [Km/s]} \\
0 \text{ [Km/s]}
\end{bmatrix}
\]

and was propagated for one year (365 days). This initial state represents a very high Earth orbit with a trajectory that is bound to a region slightly beneath the Moon’s orbit. A custom full-ephemeris propagator was created in STK by going to: “Utilities” -> “Component Browser” -> “Propagators Folder” -> Selecting a propagator and clicking “Duplicate.” Double-clicking the duplicated propagator will allow the user to change any setting of that propagator. For this study the propagator used 10 bodies and modeled each body as a point mass. The bodies involved are the eight planets (Mercury, Venus, Earth, Mars, Jupiter, Saturn, Uranus, Neptune), the Sun, and the Moon. The Earth was chosen as the “central body.” A satellite was then created and the propagator “Astrogator” was used to generate the baseline trajectory. Note that the coordinate system used by STK/Astrogator was “Earth Inertial” which is based on a geocentric J2000 coordinate system. The state vector at the end of the propagation was retrieved by going to “Analysis” -> “Report” -> “Satellite” -> “Installed Styles” -> “Inertial Position Velocity.” The final state as computed by STK was

\[
X_f = \begin{bmatrix}
228638.0987 \text{ [Km]} \\
279042.4137 \text{ [Km]} \\
20770.87224 \text{ [Km]} \\
-0.714259 \text{ [Km/s]} \\
0.107036 \text{ [Km/s]} \\
-0.010484 \text{ [Km/s]}
\end{bmatrix}
\]

If compared to the final state generated by Abraham’s Ephemeris Model, the magnitude of the difference in position is 0.1682 [Km] and the magnitude of the difference in velocity is 0.00074 [m/s]. This is an exceptionally small difference between Abraham’s Ephemeris Model and STK’s and demonstrates the high degree of accuracy and precision of Abraham’s model. For the sake of comparison Abraham’s model was re-run three times and compared with STK. Each time the model was run more and more bodies were disregarded in the calculation. The results are shown in Table 5. Note that the propagator is essentially useless after the loss of the Sun.

<table>
<thead>
<tr>
<th>Abraham’s Propagator</th>
<th>Earth + Moon + Sun</th>
<th>Earth + Moon</th>
<th>Earth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Position Error (Magnitude) [Km]</td>
<td>18.89255017</td>
<td>263892.5647</td>
<td>167715.6563</td>
</tr>
<tr>
<td>Velocity Error (Magnitude) [m/s]</td>
<td>0.040272459</td>
<td>580.9452414</td>
<td>602.8062272</td>
</tr>
</tbody>
</table>

Table 5: Propagator Error

It is important to recognize that validation of Abraham’s Ephemeris Model using only the trajectory described above may not be sufficient. The trajectories that are of primary interest in this proposal are located in the vicinity of Lagrange points. In the three body system, Lagrange points are equilibrium points. This implies that all the forces from the major players (Earth, Moon, and centripetal force) roughly sum to zero. Because of this fact, perturbations from other sources (i.e. the Sun and 7 other planets) may play an even larger role than that demonstrated in the first trajectory discussed.
above. It seems appropriate that a Libration-orbit should also be verified in Abraham’s model. Since an infinite number of Libration point orbits exist, what orbit should be chosen for validation?

To date, only one mission and two spacecraft have ever visited any Earth-Moon Lagrange points. The THEMIS (Time History of Events and Macroscale Interactions during Substorms) mission was launched in 2007 and were originally intended to study the interactions between the Sun and the Earth’s magnetic field. The mission consisted of five spacecraft and was originally intended to be terminated at the end of 2010. It was then discovered that two of the original spacecraft had just enough fuel to insert themselves into Lissajous orbits about Earth-Moon $L_1$ and $L_2$ [25, 21]. THEMIS B was re-named ARTEMIS P1 and THEMIS C was re-named ARTEMIS P2 [4]. The ARTEMIS (Acceleration, Reconnection, Turbulence and Electrodynamics of Moon’s Interaction with the Sun) mission then collected data in the vicinity of the Lagrange points for a few months. This data is available to the public and can be accessed on UC Berkeley’s website [5].

The second trajectory propagated by both STK and Abraham’s model is based on the state vector of ARTIMIS P1 (THEMIS B) on May 1, 2011 00:00:00.000 UTC. This state vector

\[
X_0 = \begin{bmatrix}
-346924.683095239 \text{ [Km]} \\
120980.805483469 \text{ [Km]} \\
16050.3697336028 \text{ [Km]} \\
-0.372332927758167 \text{ [Km/s]} \\
-0.882566329832206 \text{ [Km/s]} \\
-0.408755815977972 \text{ [Km/s]}
\end{bmatrix}
\]

was propagated for exactly 12 days, ending on May 13, 2011 00:00:00.000 UTC. Unfortunately the propagation time could not be extended beyond 12 days because the initial state vector was slightly off in its targeting and only allowed for the propagation of $3/4$ of an orbit before escaping the vicinity of the Lagrange point. The ARTIMIS spacecraft would typically perform a thruster burn every day or two in order to correct its trajectory and keep it on a proper Lissajous orbit. For that reason, it was impossible for the author to compare a later state vector with one predicted by either STK or Abraham’s model (since neither modeled thruster dynamics). Nevertheless a direct comparison between STK and Abraham’s model was performed. The magnitude of the difference in position was around $2.6 \text{ [m]}$ while the magnitude of the velocity difference is around $10^{-5} \text{ [m/s]}$. These small values indicate that over relevant propagation times, (a few months) Abraham’s model has been validated against STK. The trajectory of the spacecraft can be seen in CR3BP coordinates in Figure 10.

![Figure 10: Propagated Trajectory of ARTIMIS P1: From Left to Right: Earth, $L_1$, Moon](image_url)
and the same trajectory can also be plotted in a geocentric J2000 coordinate frame shown in Figure 11.

![Figure 11: Propagated Trajectory of ARTIMIS P1: Earth = Green, Moon = Red, Spacecraft = Blue](image)

6 Proposal to Continue Work

The existing work of Mr. Abraham has been discussed in the previous section. Building upon the work begun in the previous section, this section will focus on new work that needs to be undertaken by Mr. Abraham for a successful dissertation.

6.1 CR3BP and Optimization

6.1.1 Linearized Equations of Motion as First Guess

In Section 5 it was stated that any shooting algorithm needs to begin with an initial guess. The algorithms in Section 5 first began with a guess from tabulated values in [26] then moved to the CR3BP, and finally to the full ephemeris model. It would be far more desirable to analytically or numerically calculate the first guess rather than rely on published tables. This is where linearized equations of motion with coordinate systems centered on a Lagrange point become highly valuable. A thorough study of [8] should allow Mr. Abraham to learn and apply approximate analytic solutions of Libration point orbits.

6.1.2 More Continuation of Libration Orbit Families

Section 5 outlined some preliminary work done on Pseudo-Arclength Continuation with the goal of automatically generating multiple families of Lagrange point trajectories from an initial “seed” trajectory. This work should be further developed to account for trajectories that are outside the xy-plane as well as trajectories formed about other collinear Lagrange points ($L_2$ and $L_3$).

6.1.3 Multiple Shooting w/ Full Ephemeris

A long-term goal of this dissertation is to merge optimization algorithms with a full-ephemeris force model. This will be accomplished in a series of steps:
1. Patch points are chosen to mimic the desired trajectory (via published tables or by linearized EOMs).

2. A multiple-shooting algorithm is used with a CR3BP force model to obtain a baseline trajectory from the original patch points.

3. Since CR3BP dynamics are too crude to accurately reflect real-world conditions, a new set of patch points will be chosen from the CR3BP baseline trajectory and used to “shoot” a high-fidelity trajectory using a full-ephemeris force model (i.e. SPICE/Mice).

To date, Item 2 has been accomplished but much work is needed on Item 3. Specifically, the multiple shooting algorithm that was successfully applied to the CR3BP must now be implemented in the full-ephemeris model.

6.1.4 Monodromy Matrix, Stability & Eigenvalues, and Stable/Unstable Manifolds

The Monodromy Matrix, $M$, is associated with a given periodic, Libration point orbit. It is calculated by integrating the STM, $\Phi(t, t_0)$, for one orbital period, $P$. Therefore

$$M = \Phi(t_0 + P, t_0)$$  \hspace{1cm} (35)

Orbital stability can be assessed by looking at the eigenvalues, $\lambda$, of the Monodromy Matrix. The eigenvalues always exist in reciprocal pairs and are categorized as follows [21]:

- $||\lambda|| < 1 \rightarrow$ Stable Orbit
- $||\lambda|| = 1 \rightarrow$ Neutrally Stable Orbit
- $||\lambda|| = 1$ and $\lambda$ is real $\rightarrow$ Periodic Orbit
- $||\lambda|| > 1 \rightarrow$ Unstable Orbit

This can be illustrated in Figure [12].
If the Monodromy Matrix is generated at an arbitrary point in the spacecraft’s orbit, eigenvalues and eigenvectors can be assigned to that point. Using the eigenvectors, \( \nu \), of the local Monodromy Matrix and their associated eigenvalues allows for the identification of local stable and unstable manifolds. An unstable eigenvalue/eigenvector allows a programmer to generate the unstable manifold by slightly perturbing the spacecraft in the direction of the eigenvector and then propagating the resulting trajectory forward in time. The resulting trajectory should cause the spacecraft to depart from its orbit and follow the unstable manifold to a new destination (i.e. the Moon or another Lagrange point). Conversely, a stable manifold can be identified in the same manner except that the trajectory must be propagated backwards in time to locate this manifold. With the knowledge of this manifold in hand, a computer programmer could attempt to “patch” a trajectory segment (such as a low-thrust segment) to the beginning of the stable manifold and know that, once on the stable manifold, the spacecraft will enter the desired Libration point orbit without any further thrusting.

### 6.2 Low Thrust Trajectories

To date, Mr. Abraham has not investigated the dynamics of low thrust trajectories in detail. Since low thrust trajectories are critical to this proposal a number of goals are outlined to increase Mr. Abraham’s experience with this topic.

#### 6.2.1 Low Thrust Dynamics

Low thrust trajectories are non-Keplarian in nature and are characterized by a long-duration thrusting of a (relatively) low thrust spacecraft engine. This is in stark contrast with a high impulse (high thrust and short duration) chemical (usually liquid or solid fuel mixed with an oxidizer) maneuver that is commonplace in the dynamics of today’s spacecraft. This study is primarily interested in the low thrust dynamics of a spacecraft in LEO (Low Earth Orbit) and higher. Such trajectories appear to be spiral-shaped and increase in altitude as energy is added to the orbit.

Many outside factors can influence low thrust trajectories and will need to be modeled to propagate an accurate trajectory. In addition to the orbital perturbation generated from the engine, other forces such as atmospheric drag, the oblateness of the Earth \( J_2 \), 3rd body effects, and solar radiation pressure will have a non-trivial affect that must be accounted for. Because the majority of low thrust engines are solar-electric it is also highly important to model any eclipsing that the spacecraft will encounter since solar power will be lost (as well as thrust) during these events.

#### Figure 13: Two Examples of Low-Thrust, Spiral Trajectories (Geocentric) Taken From [12]

#### 6.2.2 Optimal Low Thrust Trajectories

Of course it is not simply enough to blindly turn on a low thrust engine and “hope for the best.” Instead one must find a way to “target” the trajectory to satisfy a particular state at a given time. Additionally, one would want to find a way to target this state while minimizing constraints on the system such as time of flight, fuel consumption, thrust limitations, or...
some combination of them all. While many low thrust optimization programs exist, it is unknown if any are suitable for finding an optimal trajectory with an end state that matches the initial state of a stable manifold of some desired Lagrange point orbit. Many of the optimization algorithms assume a two-body force model which would be highly inaccurate in a regime dominated by three-body affects.

6.2.3 Interface OLTT with Dynamical Systems (Manifold) Theory

The crux of the dissertation will be the development of a method that can successfully merge low thrust trajectories with Manifold Theory and the CR3BP. As stated previously, the goal of this dissertation is to develop a repeatable and robust algorithm that allows a geocentric, low thrust, spiral trajectory to successfully enter a Lagrange point orbit. For the sake of repeatability and robustness, the majority of the spiral trajectory will be computed without any regard to the desired Lagrange point orbit since this segment is highly sensitive to initial conditions and numerical uncertainties.

![Mission Concept: Low-Thrust Spiral (Orange) to Low-Thrust Transfer (Green) Via Stable Manifold (Blue) to Earth-Moon L1 Orbit. Image Credit [19].](image)

6.3 Summary Of Algorithm

A brief outline of the algorithm is as follows:

1. Define and find a desired Lagrange point orbit and propagate it in the $n$-body model
2. Generate the stable manifold and locate a suitable patch-point at the beginning of the manifold
3. Using a low thrust force model, propagate either towards the manifold patch-point, or start at the patch-point and propagate away from it. A combination of the two techniques may also be useful.
4. Propagate the trajectory backward to a LEO orbit

5. Pareto Efficiency & Trade Space

7 Time-line

A proposed time-line is outlined below and is based on the Proposal to Continue Work found in Section 6.

- Create a multiple-shooting algorithm in a full-ephemeris model – December, 2012
- Investigate Monodromy Matrix and create a program to automatically generate the stable manifold of an arbitrary Libration point orbit – February, 2013
- Become acquainted with low-thrust dynamics by writing some simple test programs – March, 2013
- Become familiar with low-thrust optimization schemes and demonstrate them – May, 2013
- Develop algorithm that will shoot a low-thrust trajectory to a “patch point” on a stable manifold of a desired Lagrange point orbit – July, 2013
- Begin writing dissertation – August, 2013
- Test new algorithm and generate data for dissertation – October, 2013
- Fully concentrate on dissertation after October, 2013. Stop all other work.
- Finish writing dissertation – February, 2014
- Edit dissertation – March, 2014
- Defend dissertation – April, 2014
- Graduate – May, 2014

The outline above represents the primary goals of Mr. Abraham’s dissertation. If time permits, secondary goals include:
- using linearized equations of motion as a first guess when generating an initial Lagrange point orbit, using continuation techniques to generate multiple Lagrange point orbital families from an initial “seed” orbit, and increasing the fidelity of the trajectory propagator when in a LEO orbit by incorporating atmospheric force models as well as $J_2$ perturbations.

8 Summary

In this proposal, an argument was made for the need of a robust algorithm that was capable of rapidly producing low thrust trajectories from a geocentric spiral to an Earth-Moon Lagrange point orbit. Mr. Abraham’s doctoral qualifications were outlined as well as the current progress he has made in his initial research efforts. A list of future research was presented as well as a time-line of milestones related to that research. Finally, a comprehensive summary of the theory involved in this dissertation was given in the main body of this document with generalized, introductory material relegated to the Appendix. Mr. Abraham requests that the Ph.D. committee allow him to continue his dissertation work by approving this proposal and granting him the status of Ph.D. Candidate.
References


Part II
Appendix: The Circular Restricted 3-Body Problem

Author: Andrew Abraham

www.andrewabraham.net

9 Motivation

The 3-body problem of Astrodynamics was first studied by Newton many centuries ago. Over the centuries people such as Lagrange, Euler, Hill, and Jacobi (to name a few) worked on this problem to better understand the motions of celestial objects that move under the influence of more than one object (i.e. the Moon). With the advent of the space age, the focus of the 3-body problem shifted from natural celestial bodies to artificial ones.

Early in the space age, the dominant method of trajectory calculation and mission planning was the 2-body problem. The 2-body problem was fairly simple to work with; yielding nice, algebraic trajectory solutions derived from conic sections (circle, ellipse, parabola, or hyperbola). This 2-body approximation of spacecraft dynamics was a good method to use for mission planning, but it was not without its flaws. The major drawback in the 2-body problem was that it ignores the gravitational effects of all the other objects in the solar system. As a matter of fact, some systems (such as the Earth-Moon system) cannot be accurately modeled without using 3-body dynamics. Mission planning and trajectory optimization using 3-body dynamics have become increasingly popular over the past decade with spacecraft such as: the James Webb Space Telescope, Solar and Heliospheric Observatory (SOHO), Wilkinson Microwave Anisotropy Probe (WMAP), and GRAIL operating at or near Lagrange points.

9.1 Jacobi Energy Integral

Now that we have found the equations of motion we can look at the only known conserved quantity in the CR3BP; the Jacobi Energy. Unlike the two-body problem where both energy and angular momentum are conserved, the three-body problem will conserve only the energy of the system known as the Jacobi Energy, Jacobi Integral, or Jacobi Constant. To begin this analysis, we will first look at the time derivative of the square of the velocity:

$$\frac{d}{dt}v^2 = \frac{d}{dt}(\dot{x}^2 + \dot{y}^2 + \dot{z}^2) = 2(\dot{x}\ddot{x} + \dot{y}\ddot{y} + \dot{z}\ddot{z})$$

Fortunately, we know the equations of motion for this system. By substituting equations 5 we obtain:

$$\frac{d}{dt}v^2 = 2(\dot{x}\left(2\dot{y} - \frac{\partial U}{\partial x}\right) + \dot{y}\left(-2\dot{x} - \frac{\partial U}{\partial y}\right) + \dot{z}\left(-\frac{\partial U}{\partial z}\right)) = 2\left(\ddot{x}\left(-\frac{\partial U}{\partial x}\right) + \ddot{y}\left(-\frac{\partial U}{\partial y}\right) + \ddot{z}\left(-\frac{\partial U}{\partial z}\right)\right)$$

Now recall the fact that:

$$\frac{dU}{dt} = \dot{x}\left(\frac{\partial U}{\partial x}\right) + \dot{y}\left(\frac{\partial U}{\partial y}\right) + \dot{z}\left(\frac{\partial U}{\partial z}\right)$$

which simplifies our expression to:

$$\frac{d}{dt}v^2 = -2\frac{dU}{dt} \quad \rightarrow \quad \frac{d}{dt}[-v^2 - 2U] = 0$$

32
Integrating with respect to time gives:

\[-v^2 - 2U = J\]

The constant \(J\) is known as the Jacobi Constant and represents the only known constant of the CR3BP. It can be seen that \(C\) is related to the total energy per unit mass (as measured in the rotating frame), \(\varepsilon\) by:

\[\varepsilon = \frac{1}{2}v^2 + U = \frac{1}{2}(-v^2 - 2U) = -\frac{1}{2}J \implies J = -2\varepsilon\]

One final note concerning the Jacobi Energy centers on the fact that the square of the velocity cannot, by definition, be negative. If it were negative then the velocity would have to be imaginary (and we know that doesn’t make any physical sense). If we launched a projectile (at a speed less than the escape speed) in a rectilinear trajectory from the surface of one of the primaries and into space, then eventually that projectile would slow down, stop, and then turn around. The farthest distance accessible to the projectile would occur when its velocity was exactly zero. Any region of space beyond that point is forbidden by the physics of the system (i.e. the system’s energy). It, therefore, becomes possible to map both the accessible and forbidden regions of any third-body if given its associated Jacobi Energy by the relation:

\[J = -2U_{\text{max}}\]

This equation implicitly defines an energy surface or energy manifold; no body with a given energy value can escape this manifold. Historically, the manifold of accessible space is known as “Hill’s Region” with the boundary of this region known as the “zero velocity curve.”

10 Equilibrium Points

Now we will look for the stationary points of the system. This occurs whenever \(\ddot{x} = \dot{\dot{x}} = \ddot{y} = \dot{\dot{y}} = 0\). Equation [5] becomes:

\[0 = -\frac{\partial U}{\partial x} = \frac{(\mu - 1)(x + \mu)}{\left[(x + \mu)^2 + y^2 + z^2\right]^{\frac{3}{2}}} - \frac{\mu(x + \mu - 1)}{\left[(x + \mu - 1)^2 + y^2 + z^2\right]^{\frac{3}{2}}} + x\]  

(36)

\[0 = -\frac{\partial U}{\partial y} = \frac{(\mu - 1)y}{\left[(x + \mu)^2 + y^2 + z^2\right]^{\frac{3}{2}}} - \frac{\mu y}{\left[(x + \mu - 1)^2 + y^2 + z^2\right]^{\frac{3}{2}}} + y\]  

(37)

\[0 = -\frac{\partial U}{\partial z} = \frac{(\mu - 1)z}{\left[(x + \mu)^2 + y^2 + z^2\right]^{\frac{3}{2}}} - \frac{\mu z}{\left[(x + \mu - 1)^2 + y^2 + z^2\right]^{\frac{3}{2}}} + z\]  

(38)

We note that Equation [37] and [38] are immediately satisfied anytime \(y = 0\) and \(z = 0\), respectively. Thus, we will begin exploring the existence of any stationary point which lies upon the x-axis. When \(y = z = 0\), Equation [36] simplifies to:

\[x + \frac{(\mu - 1)(x + \mu)}{\|x + \mu\|^3} - \frac{\mu(x + \mu - 1)}{\|x + \mu - 1\|^3} = 0\]  

(39)

It is very important to note that the absolute value signs in Equation [39] must be present as they are a direct consequence of the squaring of the terms in the denominator. But how can one mathematically deal with the absolute value in Equation [39]? The solution is to break up Equation [39] into three different regions based on the magnitudes of the terms within the absolute value operation.
1. Region I consists of the section of the x-axis between Primary 1 and Primary 2 where \( L_1 \) is defined by \(-\mu < L_1 < 1 - \mu\). Note that \( \|L_1 + \mu\| = L_1 + \mu \) and \( \|L_1 + \mu - 1\| = -(L_1 + \mu - 1) \). Substituting this into Equation 39 gives:

\[
L_1 (L_1 + \mu)^2 (L_1 + \mu - 1)^2 + (\mu - 1) (L_1 + \mu - 1)^2 + \mu (L_1 + \mu)^2 = 0
\] (40)

If you expand Equation 40 you will get a 5th order polynomial in terms of \( L_1 \). Unfortunately, this polynomial can not be solved for algebraically; only a numerical solution using a numeric value for \( \mu \) can be found. The numerical solution will yield 5 roots; one of which is real. This real-root is the location of the stationary point corresponding to \( L_1 \). In the case of the Earth-Moon system the position of \( L_1 = 0.8369 \) [du].

2. Region II consist of the x-axis lying to the right of Primary 2 where \( L_2 \) is defined by \(-\mu < 1 - \mu < L_2\). Note that \( \|L_2 + \mu\| = L_2 + \mu \) and \( \|L_2 + \mu - 1\| = L_2 + \mu - 1 \). Substituting this into Equation 39 gives:

\[
L_2 (L_2 + \mu)^2 (L_2 + \mu - 1)^2 + (\mu - 1) (L_2 + \mu - 1)^2 - \mu (L_2 + \mu)^2 = 0
\] (41)

Again, this equation is a 5th order polynomial, contains only one real-root, and can only be solved for numerically. In the case of the Earth-Moon system the position of \( L_2 = 1.1557 \) [du].

3. We will call the stationary point within Region III \( L_3 \). This region will lie to the left of Primary 1 and is defined by \(-1 - \mu < \mu < L_3\). Note that \( \|L_3 + \mu\| = -(L_3 + \mu) \) and \( \|L_3 + \mu - 1\| = -(L_3 + \mu - 1) \). Substituting this into Equation 39 gives:

\[
L_3 (L_3 + \mu)^2 (L_3 + \mu - 1)^2 - (\mu - 1) (L_3 + \mu - 1)^2 + \mu (L_3 + \mu)^2 = 0
\] (42)

As before, this equation is another 5th order polynomial, contains only one real-root, and can only be solved for numerically. In the case of the Earth-Moon system the position of \( L_3 = -1.00506 \) [du].

In this way three Lagrange points, \( L_1, L_2, \) and \( L_3 \) can be found when \( y = z = 0 \). Now we will attempt to find any remaining Lagrange points when \( y \neq 0 \). The solution is actually intractable without taking advantage of the symmetry of the system. Please refer to Equation 36. This equation is actually very difficult to deal with unless the values of the two denominators were equal to each other (and this is the key to making the problem tractable). By inspection, you will note that the values \( x = \frac{1}{2} - \mu, z = 0 \) would be an appropriate substitution that allows for further simplification of Equation 36 and Equation 37 to \( y = \pm \sqrt{\frac{1}{4}} \). Thus the remaining two Lagrange points are symmetric and have values of \( L_4 = \left( \frac{1}{2} - \mu, \sqrt{\frac{1}{4}} \right) \) and \( L_5 = \left( \frac{1}{2} - \mu, -\sqrt{\frac{1}{4}} \right) \).

10.1 Jacobi Energy at Equilibrium Points

Interestingly enough, we can calculate the Jacobi Energy of a particle at rest (in the co-rotating system) and positioned at any of the five Lagrange Points. Below is a table of the Lagrange Points for the Earth-Moon system:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>( L_1 )</td>
<td>0.8369</td>
<td>0</td>
<td>0</td>
<td>saddle</td>
<td>3.1883</td>
</tr>
<tr>
<td>( L_2 )</td>
<td>1.1557</td>
<td>0</td>
<td>0</td>
<td>saddle</td>
<td>3.1722</td>
</tr>
<tr>
<td>( L_3 )</td>
<td>-1.00506</td>
<td>0</td>
<td>0</td>
<td>saddle</td>
<td>3.0121</td>
</tr>
<tr>
<td>( L_4 )</td>
<td>0.487846</td>
<td>0.86602</td>
<td>0</td>
<td>stable</td>
<td>2.9879</td>
</tr>
<tr>
<td>( L_5 )</td>
<td>0.487846</td>
<td>-0.86602</td>
<td>0</td>
<td>stable</td>
<td>2.9879</td>
</tr>
</tbody>
</table>

Correspondingly, below is a series of graphs that plot the Jacobi Energies \( J_1-J_{4,5} \) for their corresponding Lagrange Points:

34
The black area in the figure above represents the “forbidden region” in which the third body is forbidden to exist given a fixed amount of energy. At energies below $J_1$ there are three disconnected realms: the Earth-Centered Realm, the Moon-Centered Realm, and the Exterior Realm. Once the energy is equal to that of $J_1$ the Earth and Moon realms merge at the location of $L_1$; it now becomes possible for the third body to move from the Earth realm to the Moon realm (and visa versa) through the “neck” region located around $L_1$. As the energy increases further, to that of $L_2$, the Earth, Moon, and Exterior realms all become connected. At this energy level it becomes possible for the third body to freely move between all realms by traveling through the $L_1$-Moon-$L_2$ neck region. As the energy is further increased, to that of $L_3$, another neck opens in the opposite direction. Finally, once the energy approaches $J_{4,5}$ all of the “forbidden” regions cease to exist.

11 Linearization, Stability, and Bifurcation

Now that the positions of the Lagrange Points (i.e. equilibrium points) are known we must find a way to characterize
their stability. To accomplish this stability caricatureization, we must define the state vector of the third body as:

\[ X = \begin{bmatrix} p \\ q \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} \]

and the time derivative of the state vector is:

\[ \dot{X} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \tilde{f}(X) = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ 2\dot{y} - \frac{\partial U}{\partial y} \\ -2\dot{x} - \frac{\partial U}{\partial x} \\ -\frac{\partial U}{\partial z} \end{bmatrix} \]

The Jacobian Matrix can be written as:

\[
A(t) = \begin{bmatrix}
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
-\frac{\partial^2 U}{\partial x^2} & -\frac{\partial^2 U}{\partial xy} & -\frac{\partial^2 U}{\partial xz} & -\frac{\partial^2 U}{\partial x^2} & 2 & -\frac{\partial U}{\partial x} \\
-\frac{\partial^2 U}{\partial y^2} & -\frac{\partial^2 U}{\partial yx} & -\frac{\partial^2 U}{\partial yz} & -\frac{\partial^2 U}{\partial y^2} & 0 & 2 \\
-\frac{\partial^2 U}{\partial z^2} & -\frac{\partial^2 U}{\partial zx} & -\frac{\partial^2 U}{\partial zy} & -\frac{\partial^2 U}{\partial z^2} & 0 & 0 \\
\end{bmatrix}
\]

Note that:

\[
a = -\frac{\partial^2 U}{\partial x^2} \quad b = -\frac{\partial^2 U}{\partial xy} \quad c = -\frac{\partial^2 U}{\partial y^2} \quad d = -\frac{\partial^2 U}{\partial xz} \quad e = -\frac{\partial^2 U}{\partial yz} \quad f = -\frac{\partial^2 U}{\partial z^2}
\]

Expanding these terms gives:

\[
a = \frac{3(1-\mu)(x+\mu)^2}{((x+\mu)^2+y^2+z^2)^2} - \frac{1-\mu}{((x+\mu)^2+y^2+z^2)^2} + \frac{3\mu(x+\mu-1)^2}{((x+\mu-1)^2+y^2+z^2)^2} - \frac{\mu}{((x+\mu-1)^2+y^2+z^2)^2} + 1
\]

\[
b = \frac{3(1-\mu)(x+\mu)y}{((x+\mu)^2+y^2+z^2)^2} + \frac{3\mu y(x+\mu-1)}{((x+\mu-1)^2+y^2+z^2)^2}
\]

\[
c = \frac{3(1-\mu)y^2}{((x+\mu)^2+y^2+z^2)^2} - \frac{1-\mu}{((x+\mu)^2+y^2+z^2)^2} + \frac{3\mu y^2}{((x+\mu-1)^2+y^2+z^2)^2} - \frac{\mu}{((x+\mu-1)^2+y^2+z^2)^2} + 1
\]
\[ d = \frac{3(1-\mu)(x+\mu)z}{((x+\mu)^2+y^2+z^2)^2} + \frac{3\mu z (x+\mu-1)}{((x+\mu-1)^2+y^2+z^2)^2} \]
\[ e = \frac{3(1-\mu)yz}{((x+\mu)^2+y^2+z^2)^2} + \frac{3\mu yz}{((x+\mu-1)^2+y^2+z^2)^2} \]
\[ f = \frac{3(1-\mu)z^2}{((x+\mu)^2+y^2+z^2)^2} - \frac{1-\mu}{((x+\mu)^2+y^2+z^2)^2} + \frac{3\mu z^2}{((x+\mu-1)^2+y^2+z^2)^2} - \frac{\mu}{((x+\mu-1)^2+y^2+z^2)^2} \]

### 11.1 Co-linear Libration Point Stability

Unfortunately, the only way to continue our study of the co-linear stability points is through numerical computation with a given value for \( \mu \). Upon numerical computation, we find that for the Earth-Moon system:

\[
\mu \approx 0.0121 \quad \begin{cases} 
L_1 = (0.8369,0) & \text{Unstable} \rightarrow \text{Saddle} \\
L_2 = (1.1557,0) & \text{Unstable} \rightarrow \text{Saddle} \\
L_3 = (-1.0050,0) & \text{Weakly Unstable}
\end{cases}
\]

and for the Sun-Earth system:

\[
\mu \approx 3 \times 10^{-6} \quad \begin{cases} 
L_1 = (0.9900,0) & \text{Unstable} \rightarrow \text{Weak Saddle} \\
L_2 = (1.0100,0) & \text{Unstable} \rightarrow \text{Weak Saddle} \\
L_3 = (-1.0000,0) & \text{Very Weakly Unstable}
\end{cases}
\]

### 11.2 \( L_3 \) Stability Approximation with Low Values of \( \mu \)

In the 19th century it was believed that a hidden planet, known as Planet X, may be hiding behind the Sun at the Sun-Earth \( L_3 \) point. They feared that this planet was inhabited by intelligent creatures who desired to travel to Earth and take over the planet. The fact that \( L_3 \) is continuously hidden from Earth’s view by the Sun terrified some scientists of the day. We can make some useful approximations of the stability dynamics of the \( L_3 \) point if we assume that \( \mu \ll \frac{1}{2} \); say \( \mu \leq 1\% \). In this case, the x coordinate of the \( L_3 \) point is very well approximated by \( x \approx -1 \) (and, of course, \( y = z = 0 \)). Plugging this into the expressions for a-f gives:

\[
\begin{align*}
    a &= \frac{-24+50\mu-46\mu^2+25\mu^3-8\mu^4+\mu^5}{-8+2\mu-15\mu^2+2\mu^3-8\mu^4+\mu^5} \\
    b &= 0 \\
    c &= \frac{17\mu-34\mu^2+25\mu^3-8\mu^4+\mu^5}{-8+2\mu-15\mu^2+2\mu^3-8\mu^4+\mu^5} \\
    d &= 0 \\
    e &= 0 \\
    f &= \frac{8-11\mu+4\mu^2}{-8+2\mu-15\mu^2+2\mu^3-8\mu^4+\mu^5}
\end{align*}
\]
Since $\mu$ is small we can neglect all but the lowest term in each expression:

$$
\begin{align*}
a &= 3 \\
b &= 0 \\
c &= -\frac{17\mu}{8} \\
d &= 0 \\
e &= 0 \\
f &= -1
\end{align*}
$$

Now the Jacobian Matrix becomes:

$$
Df = \begin{bmatrix}
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 \\
3 & 0 & 0 & 0 & 2 & 0 \\
-\frac{17\mu}{8} & 0 & -2 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & 0
\end{bmatrix}
$$

with corresponding eigenvalues of:

$$
\lambda_{L_3, \mu \ll \frac{1}{2}} = \begin{bmatrix}
i \\
-i \\
\frac{1}{4} \sqrt{-8 - 17\mu + \sqrt{64 + 1904\mu + 289\mu^2}} \\
-\frac{1}{4} \sqrt{-8 - 17\mu + \sqrt{64 + 1904\mu + 289\mu^2}} \\
\frac{1}{4} \sqrt{-8 - 17\mu - \sqrt{64 + 1904\mu + 289\mu^2}} \\
-\frac{1}{4} \sqrt{-8 - 17\mu - \sqrt{64 + 1904\mu + 289\mu^2}}
\end{bmatrix}
$$

We will immediately neglect the $\mu^2$ term since it is very small compared with the other terms. Next we can expand the inner square root term by using the binomial expansion:

$$
(1 + x)^r = \sum_{k=0}^{\infty} \frac{(-r)^k}{k!} (-x)^k = 1 + rx + \frac{1}{2} r(r+1) x^2 + \frac{1}{6} r(r+1)(r+2) x^3 + ...
$$

$$
\sqrt{64 + 1904\mu} = 8 \left( 1 + \frac{119}{4} \mu \right)^{\frac{1}{2}} = 8 \left[ 1 + \frac{119}{8} \mu + \frac{1}{8} \left( \frac{119}{8} \mu \right)^2 + ... \right] \approx 8 + 119\mu
$$

Substituting this value back into the expression for the eigenvalues (and noting that $0 \leq \mu \leq \frac{1}{2}$) gives:

$$
\lambda_{L_3, \mu \ll \frac{1}{2}} = \begin{bmatrix}
i \\
-i \\
\frac{1}{4} \sqrt{102\mu} \\
-\frac{1}{4} \sqrt{102\mu} \\
\frac{1}{4} \sqrt{-16 - 136\mu} \\
-\frac{1}{4} \sqrt{-16 - 136\mu}
\end{bmatrix} \approx \begin{bmatrix}
i \\
-i \\
\frac{\sqrt{17}}{2} \sqrt{\mu} \\
-\frac{\sqrt{17}}{2} \sqrt{\mu} \\
\sqrt{1 + \frac{17}{2} \mu} i \\
-\sqrt{1 + \frac{17}{2} \mu} i
\end{bmatrix}
$$
Note that we have one positive, real eigenvalue that is a function of the square root of $\mu$. The $L_3$ point is, indeed, unstable but the amount of instability has now been directly expressed in terms of $\mu$ when $\mu \ll \frac{1}{2}$. The motion of a body at this $L_3$ point will diverge as a function of:

$$e^{\lambda_{L_3} t} \approx e^{\frac{1}{2} \sqrt{\mu} t}$$

The time-constant $\tau$ can be defined as $\tau = \frac{1}{\lambda} = \frac{2}{\sqrt{\mu}}$. Since $\mu \ll \frac{1}{2}$ we know that $\tau$ is relatively large. Indeed for the Sun-Earth system $\tau \approx 230$ Time Units (years in the Sun-Earth system). While Planet X can’t exist at the $L_3$ point (due to its instability), intelligent invaders from outer space could still park a hidden Earth-invasion, space-fleet behind the Sun and trust that their first spaceship would still be there more than 230 years later.

### 11.3 $L_4$ and $L_5$ Stability and Bifurcation

The $L_4$ and $L_5$ points, on the other hand, can be solved for algebraically. The values of the matrix components $a$, $b$, and $c$ are:

**for $L_4$:** $a = \frac{3}{4}$, $b = \frac{3}{4} \sqrt{3} - \frac{3}{2} \sqrt{3} \mu$  
$c = \frac{9}{4}$  
$d = 0$  
$e = 0$  
$f = -1$

**for $L_5$:** $a = \frac{3}{4}$, $b = \frac{3}{4} \sqrt{3} + \frac{3}{2} \sqrt{3} \mu$  
$c = \frac{9}{4}$  
$d = 0$  
$e = 0$  
$f = -1$

In either case, the eigenvalues for both $L_4$ and $L_5$ turn out to be exactly the same:

$$\lambda_{L_4,5} = \begin{cases} 
& i \\
& -i \\
& \frac{1}{2} \sqrt{-2 + 2 \sqrt{1 - 27 \mu + 27 \mu^2}} \\
& -\frac{1}{2} \sqrt{-2 + 2 \sqrt{1 - 27 \mu + 27 \mu^2}} \\
& \frac{1}{2} \sqrt{-2 - 2 \sqrt{1 - 27 \mu + 27 \mu^2}} \\
& -\frac{1}{2} \sqrt{-2 - 2 \sqrt{1 - 27 \mu + 27 \mu^2}} 
\end{cases}$$

It is now helpful to define a new value, $k = \frac{m_1 - m_2}{m_1 + m_2} = 1 - 2\mu$, to simplify the eigenvalues. Note that the range of $k$ is between 0 and 1. Upon substitution we have:

$$\lambda_{L_4,5} = \begin{cases} 
& i \\
& -i \\
& \frac{1}{2} \sqrt{-2 + \sqrt{27} k^2 - 23} \\
& -\frac{1}{2} \sqrt{-2 + \sqrt{27} k^2 - 23} \\
& \frac{1}{2} \sqrt{-2 - \sqrt{27} k^2 - 23} \\
& -\frac{1}{2} \sqrt{-2 - \sqrt{27} k^2 - 23} 
\end{cases}$$

For an equilibrium point to be stable (or at least not unstable), the real part of each eigenvalue of the Jacobian Matrix must be non-positive. Looking at the eigenvalues above, it becomes apparent that they need to become entirely imaginary to satisfy this condition; if the eigenvalues had any non-zero real part then half would be negative and half would be positive which obviously breaks the stability condition. By inspection, we note that the following conditions must be met for all eigenvalues to be imaginary:

$$\begin{cases} 
\sqrt{27} k^2 - 23 \leq 2 & (1) \\
27 k^2 - 23 \geq 0 & (2) 
\end{cases} = \begin{cases} 
k \leq \frac{1}{\sqrt{\frac{23}{27}}} & (1) \\
k \geq \sqrt{\frac{23}{27}} & (2) 
\end{cases}$$
Since the range of $k$ is between zero and one; condition (1) is always satisfied. Condition (2) implies that $0 \leq k \leq 0.9229$ or $0 \leq \mu \leq 0.0385$ or $0 \leq \frac{m_2}{m_1} \leq 4\%$ (since $\frac{m_2}{m_1} = \frac{\mu}{1-\mu}$). This is a remarkable result. The stability of the $L_4$ and $L_5$ equilibrium points is a function of $\mu$ only. The system will bifurcate, going from stable to unstable, once the critical value of $\mu_C = \frac{1}{2} - \frac{\sqrt{18}}{18} \approx 0.0385$ has been crossed.