

## ABSTRACT

Title of Thesis: INVESTIGATION OF MANIFOLDS AND  
OPTIMIZED TRAJECTORIES IN THE  
THREE-BODY PROBLEM

**Raquel L. Jarabek, M.S., 2004**

Thesis Directed By: **Assistant Professor Benjamin Shapiro**  
**Department of Aerospace Engineering**

Manifolds and optimal control were used to better understand trajectories in the circular restricted three-body problem (CR3BP). CR3BP equations were used to generate two-dimensional stable and unstable manifolds. Optimized trajectory solutions were found using the Hamilton-Jacobi-Bellman equation applied to the third body traveling from  $L_1$  toward  $m_2$ . Three sets of optimal trajectories with various fixed final positions were compared to the  $L_1$  manifold. The cases with final positions closest to the manifold remained close and had lower costs. Trajectories with low time allowances took more direct paths to their final positions, leaving the manifold and resulting in higher costs. Large time allowances caused increased trajectory length and early departure from the manifold, resulting in increased cost. For the intermediate time constrained cases, the trajectories stay longer on the manifold and cost less. From this investigation, optimized trajectories were shown to use manifolds when finding the optimal trajectory in the CR3BP.

INVESTIGATION OF MANIFOLDS AND OPTIMIZED TRAJECTORIES IN THE  
THREE-BODY PROBLEM

By

Raquel L. Jarabek

Thesis submitted to the Faculty of the Graduate School of the  
University of Maryland, College Park, in partial fulfillment  
of the requirements for the degree of  
Master of Science in  
Aerospace Engineering  
2004

Advisory Committee:  
Assistant Professor Benjamin Shapiro, Chair  
Associate Professor David L. Akin  
Associate Professor Robert Sanner

© Copyright by  
Raquel L. Jarabek  
2004

## Dedication

This thesis is dedicated to God for helping me achieve the goal of finishing my masters. Without Him, I would not have had the strength or knowledge to complete this project. It is because of God that I am on this path and hoping to achieve much more in the field of aerospace engineering.

## Acknowledgements

First and foremost, I need to thank my husband, Bryan, for loving me unconditionally and supporting me more than I could have imagined was possible. Your love and encouragement help me to believe that I can accomplish anything. You believe in me when others don't and help pick me up when I am discouraged. Your ability to understand science and engineering constantly amazes me. The conversations we have had about my research made my project better and allowed me to complete this thesis. I cannot thank you enough for all of the help you have provided.

Thank you to Michel Santos and Dr. Ben Shapiro for all that you contributed to this research project. This project could not have been accomplished without the contributions of Michel's research. Michel, thank you for allowing me to use part of your project to accomplish mine. Ben, thank you for your patience and understanding over the last three years as my life evolved and challenges arose.

Thank you to Greg Marr and Tom Stengle at NASA Goddard Space Flight Center for your expertise and knowledge that you shared with me. Thanks to the NASA Graduate Student Research Program for providing the funding that made this research possible.

There are two friends with aerospace experience that I need to specifically thank for their encouragement and thesis revisions. Thank you, Rhiannon and Laurie, for all that you provided to help get me through my masters. Thank you to all of the friends and family that have loved and supported me and prayed for me throughout my graduate school experience.

# Table of Contents

Dedication.....	ii
Acknowledgements.....	iii
Table of Contents.....	iv
List of Figures.....	vi
List of Tables.....	vii
1. Introduction.....	1
2. The Three-Body Problem.....	4
3. Methods.....	9
3.1 Solving for the Lagrange Points.....	9
3.2 Establishing the Stability of the Lagrange Points.....	10
3.3 Solving for Zero Velocity Curves.....	12
3.4 Generating the Manifolds for the CR3BP.....	15
3.5 Developing the Hamilton-Jacobi-Bellman Optimization Algorithm.....	17
4. Results.....	21
4.1 Manifold Solution of the CR3BP.....	21
4.2 HJB Optimized Solution of the CR3BP.....	28
4.3 Comparison of Manifold and HJB Optimized Solutions About the $L_1$ Point .....	32
5. Conclusions.....	42
6. Appendices.....	46
6.1 Appendix A.....	46
7.2 Appendix B.....	67

6.3 <i>Appendix C</i> .....	68
6.4 <i>Appendix D</i> .....	69
7. References.....	70

## List of Figures

Figure 1. Description of the three-body problem coordinate system.....	7
Figure 2. The locations of the 5 Lagrange points in the normalized coordinate system.....	8
Figure 3. Curves of constant $C$ values and zero velocity.....	15
Figure 4. The $L_1$ manifold propagated over a long period of time. ....	23
Figure 5. The $L_1$ manifold propagated over a short period of time.....	24
Figure 6. The $L_2$ manifold.....	25
Figure 7. The $L_3$ manifold.....	26
Figure 8. The $L_4$ manifold.....	27
Figure 9. HJB optimization results from $L_1$ to $(-0.91, 0)$ .....	30
Figure 10. HJB optimization results from $L_1$ to $(-0.92, 0)$ .....	31
Figure 11. HJB optimization results from $L_1$ to $(-0.95, 0)$ .....	32
Figure 12. Comparison of the manifold to HJB optimization results from $L_1$ to $(-0.91,$ $0)$ with an unconstrained end velocity.....	34
Figure 13. Comparison of the manifold to HJB optimization results from $L_1$ to $(-0.91,$ $0)$ .....	35
Figure 14. Comparison of the manifold to HJB optimization results from $L_1$ to $(-0.92,$ $0)$ .....	38
Figure 15. Comparison of the manifold to HJB optimization results from $L_1$ to $(-0.95,$ $0)$ .....	40
Figure 16. Comparison of the normal and looped cases of HJB optimization results from 2.5TU.....	40



## List of Tables

Table 1. Results for Case 1: $x_0 = (-.609, 0, 0, 0)$ ; $x_f = (-.91, 0, -3.16, 0)$ ;	36
Table 2. Results for Case 2: $x_0 = (-.609, 0, 0, 0)$ ; $x_f = (-.92, 0, \text{unconstrained}, \text{unconstrained})$	38
Table 3. Results for Case 3: $x_0 = (-.609, 0, 0, 0)$ ; $x_f = (-.95, 0, \text{unconstrained}, \text{unconstrained})$	42

## 1. Introduction

In the eighteenth century, Joseph-Louis Lagrange and Leonard Euler better defined the restricted problem of three bodies. He realized that when the two primary bodies move in circular orbits there are five points in the bodies' plane of motion where the forces acting on each point are balanced or in equilibrium. The forces that are in equilibrium include gravitational attractions of the two large masses on the satellite and the centripetal force on the satellite causing it to move with the bodies. Lagrange and Euler discovered these points between 1665 and 1772 and today the five points are known as Lagrange points (also referred to as libration or equilibrium points). Three of the points ( $L_1$ ,  $L_2$ ,  $L_3$ ) are collinear along a line connecting the two primary bodies. They are unstable and a satellite can be kept in orbit about one of them with minimal station-keeping, because there are no large perturbing forces. The other two points ( $L_4$ ,  $L_5$ ) each form equilateral triangles with the two primary bodies and are quasi-stable for small mass ratios. If a satellite is placed at one of the triangular points, then it will librate, or oscillate around an equilibrium point if the ratio of the two larger masses is less than 0.0385. The small mass ratio constraint is met by two-body systems including: Earth-Moon, Sun-Earth, and the Sun with Mercury, Venus, Mars, Jupiter, Saturn, Uranus, Neptune, or Pluto.

Lagrange points have many uses and some satellites are already making use of Lagrange point orbits. The greatest potential advantages from using Lagrange point orbits include: decreased Earth occultation time, decreased variation in the thermal environment when compared to low Earth orbits, decreased spacecraft distance from

Earth (for Sun-Earth and Earth-Moon systems) in comparison to some heliocentric orbits, decreased station-keeping fuel requirements, and increased viewing of Earth's illuminated hemisphere (for Sun-Earth and Earth-Moon systems).

There are numerous satellites that have utilized Lagrange points. The satellites that have had Lagrange point orbits in the Sun-Earth system and the Earth-Moon system include: ISEE-3 (International Sun-Earth Explorer-3) in 1978, Wind in 1994, SOHO (Solar and Heliospheric Observatory) in 1995, ACE (Advanced Composition Explorer) in 1997, Genesis in 2001, and MAP (Microwave Anisotropy Probe) in 2001. Future satellites planning to use Lagrange point orbits include: Planck and Herschel in 2007, James Webb Space Telescope (formerly Next Generation Space Telescope) in 2010, and SPICA (SPace Infrared telescope for Cosmology and Astrophysics).

Because of the large number of current and future spacecraft missions using Lagrange points, this area of research is getting a lot of attention. Currently, the trajectory design for Lagrange point missions is not systematic but involves trial and error to find a trajectory that will accomplish the requirements of a mission. A nominal trajectory is usually based on previous solutions to the two-body problem using conic approximations, but this does not provide adequate accuracy needed for current Lagrange point missions. The trajectory design that is needed most is from a low Earth orbit to an orbit about a Lagrange point. The best solution to trajectory design

would be found in a general solution to the three-body problem, but that does not currently exist.

One way to address the three-body problem is to look at the manifolds within the dynamic system. A manifold is defined as a surface where an orbit starting on the surface will remain on that surface. (Howell et al. 1997a; Howell et al. 1994; Lo and Ross 1998) The eigenvectors define the stable and unstable manifolds. A stable manifold for the CR3BP is defined as a surface where the respective eigenvector results in motion in the direction toward its corresponding Lagrange point. An unstable manifold is defined as a surface where the dynamics of that eigenvector dictate the motion to be away from the corresponding Lagrange point. More research is being done using manifolds and dynamic systems theory to develop trajectories because they are based on the dynamics of the three-body problem and will help in better understanding how to develop trajectories using the smallest cost possible. Examples of this research include the study of the center manifold to develop halo orbits about the collinear Lagrange points (Lo 1997) and the study of dynamic systems theory as it applies to specific Lagrange point missions. (Gomez et al. 1993; Howell et al. 1997b)

Optimal control methods need to be used in conjunction with a dynamical systems approach to better prove that manifolds are the optimal way to travel to Lagrange points. Optimal control has been used to generate halo orbits around Lagrange points. (Serban et al. 2002) Zero cost transfers using optimal control for transfer

trajectories from one libration point to another have also been studied. (Gomez and Masdemont 2000) Some optimal transfers from Earth to the Earth-Moon  $L_2$  were examined by Starchville and Melton. (Starchville and Melton 1998) Despite these specific examples, optimal control remains under-utilized in the three-body problem. As additional uses for Lagrange orbits are discovered, more satellites will make use of Lagrange points. One way to better plan trajectories from Earth to Lagrange points would be to find the optimal path by investigating manifold theory to determine the stable optimal pathway. This thesis examines using optimal control and manifolds to better understand trajectories from a Lagrange point toward a primary mass in the three-body problem. Hopefully as the dynamics of the three-body problem are better understood new trajectories to Lagrange points will be discovered.

## 2. The Three-Body Problem

Euler and Lagrange first defined the circular restricted three-body problem in 1772. The three-body problem is defined as having two massive bodies and a third infinitesimal mass. The problem is described as circular because the two massive primary bodies are assumed to move in circular orbits about their common center of mass. The motion is restricted because the two large bodies are assumed to be fixed with respect to each other. (The acronym CR3BP will be used to refer to the circular restricted three-body problem.)

From Newton's laws of motion and the law of gravitation, the CR3BP can be defined.

The force on the  $i^{th}$  mass from Newton's laws for three bodies is:

$$m_i \ddot{\bar{R}}_i = G \sum_{j=1}^3 \frac{m_i m_j}{r_{ij}^3} \vec{r}_{ij}, \quad j \neq i, i = 1, 2, 3 \quad (1)$$

where  $\vec{r}_{ij} = \bar{R}_j - \bar{R}_i$ ;  $\vec{r}_{ij} = -\vec{r}_{ji}$

In Equation 1,  $\vec{r}_{ij}$  is the vector from mass  $m_i$  to  $m_j$  and  $G$  is the gravitational constant.

The radius vectors,  $\bar{R}$ , are from the origin of the coordinate system (also the center of mass of the two primaries) to each mass. Equation 1 is used to show the force on  $m_3$  by  $m_1$  and  $m_2$ .

It is helpful to introduce the standard nondimensional convention used to represent the three-body problem. First, a choice is made to define the distance between the two primaries to be one.

$$r_{12} = |\bar{R}_2 - \bar{R}_1| = 1 \quad (2)$$

The mass of the third body is infinitesimal, so choosing the mass of the two primaries to equal one is also a valid simplification.

$$m_1 + m_2 = 1 \quad (3)$$

The mass of the smaller primary can be assigned to be  $\mu$ , so that the masses are

$$\begin{aligned} m_1 &= 1 - \mu \\ m_2 &= \mu \end{aligned} \quad (4)$$

The gravitational parameter,  $\mu$ , can also be defined as a mass ratio.

$$\mu = \frac{m_2}{m_1 + m_2} \quad (5)$$

The angular velocity,  $\omega$ , of the two primaries about their center of mass is defined by Kepler's third law and the definition can be applied to the three-body problem.

$$\omega = \frac{G(m_1 + m_2)}{r_{12}^3} \quad (6)$$

Then the magnitudes of the radius vectors,  $\vec{r}_{ji}$ , can be rewritten as

$$\begin{aligned} r_1^2 &= (x - \mu)^2 + y^2 + z^2 \\ r_2^2 &= (x + 1 - \mu)^2 + y^2 + z^2 \end{aligned} \quad (7)$$

where  $r_{13} = r_1$  and  $r_{23} = r_2$ .

Equations 2 through 7 were applied to simplify Equation 1 resulting in a new representation of the CR3BP equations of motion labeled Equation 8.

$$\begin{aligned} \ddot{x} - 2\omega\dot{y} - \omega^2x &= -G\frac{m_1}{r_1^3}(x - \mu) - G\frac{m_2}{r_2^3}(x + 1 - \mu) \\ \ddot{y} + 2\omega\dot{x} - \omega^2y &= -G\frac{m_1}{r_1^3}y - G\frac{m_2}{r_2^3}y \\ \ddot{z} &= -G\frac{m_1}{r_1^3}z - G\frac{m_2}{r_2^3}z \end{aligned} \quad (8)$$

Lagrange was the first person to develop particular solutions to this set of equations of motion in 1772. Equation 8 can be further simplified by assuming the gravitational constant to be equal to one. Now, from Kepler's third law (Equation 6), the angular velocity is one.

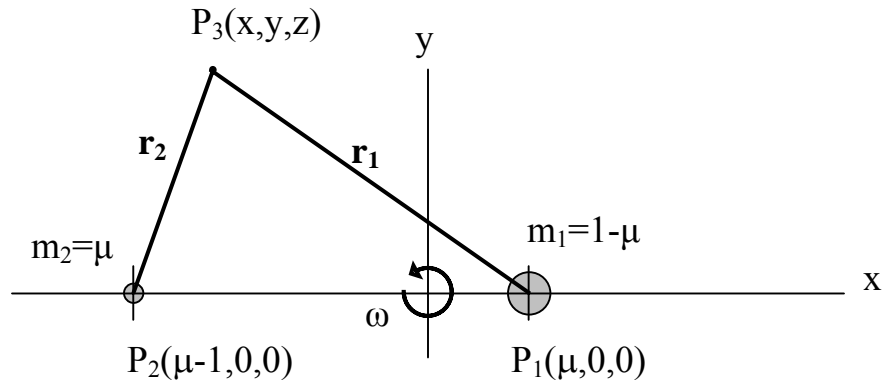
$$\begin{aligned} G &= 1 \\ \omega &= 1 \end{aligned} \tag{9}$$

After applying Equation 9 to Equation 2, the equations of motion can be written as

$$\begin{aligned} \ddot{x} - 2\dot{y} - x &= -\frac{(1-\mu)(x-\mu)}{r_1^3} - \frac{\mu(x+1-\mu)}{r_2^3} \\ \ddot{y} + 2\dot{x} - y &= -\frac{(1-\mu)y}{r_1^3} - \frac{\mu y}{r_2^3} \\ \ddot{z} &= -\frac{(1-\mu)z}{r_1^3} - \frac{\mu z}{r_2^3} \end{aligned} \tag{10}$$

Equation 10 is a set of three second-order, nonlinear, coupled ordinary differential equations that are a mathematical representation of the CR3BP. This set of equations has been normalized and the units of each parameter are in terms of length units (LU), time units (TU), and mass units (MU).

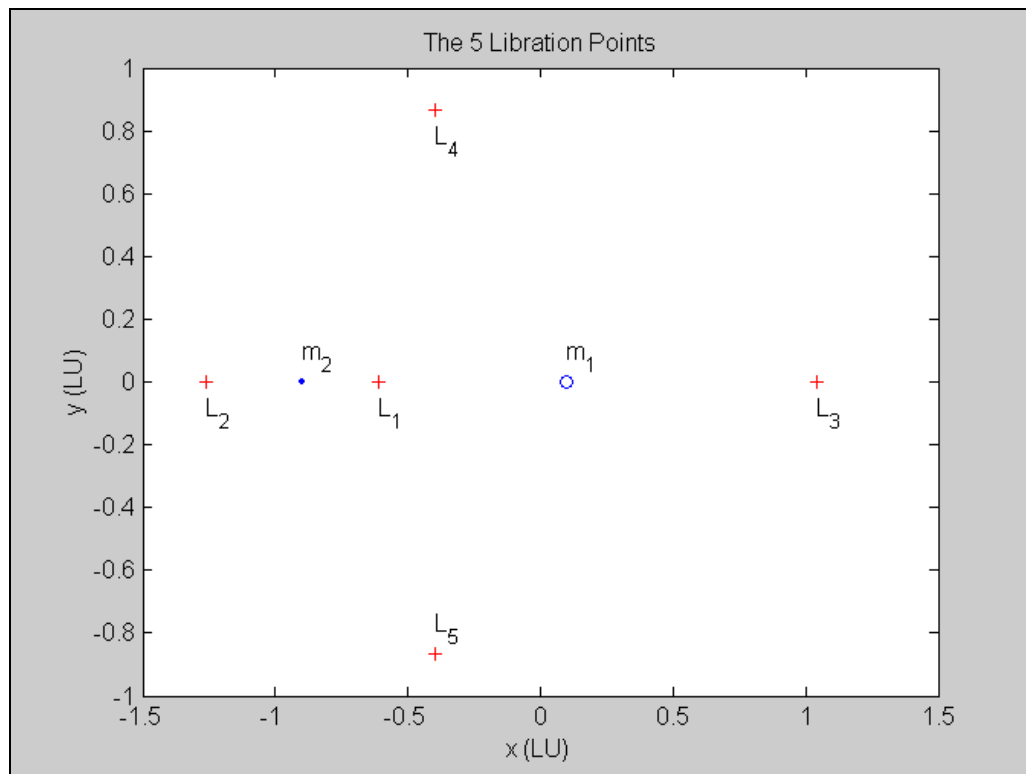
A three-body example is shown in Figure 1. The origin is placed at the center of mass of the two large bodies. The first large mass,  $m_1$ , is at a distance of  $-\mu$  from the origin. The second large mass,  $m_2$ , is at a distance of  $\mu - 1$ .



**Figure 1. Description of the three-body problem coordinate system.**



Euler and Lagrange are credited with proving the existence of five equilibrium points in the CR3BP. They discovered the five points between 1765 and 1772. They found that when the two primary bodies move on circles about their center of mass, there are five points where the forces are in equilibrium. The forces that are balanced at the five points include gravitational attractions of the two large bodies and the centrifugal force acting on the third body. The five points are known as Lagrange points, libration points, or equilibrium points and they are shown in Figure 2. (Roy 1965)



**Figure 2. The locations of the 5 Lagrange points in the normalized coordinate system.**

### 3. Methods

#### 3.1 Solving for the Lagrange Points

The five Lagrange points can be found from the CR3BP equations. In order to solve for the five points, the masses are assumed to be at rest in the rotating-frame, so the velocity and acceleration components are zero. Then Equation 10 simplifies to Equation 11 and the 5 equilibrium points can be found from the following set of equations.

$$\begin{aligned}x &= \frac{(1-\mu)(x-\mu)}{r_1^3} + \frac{\mu(x+1-\mu)}{r_2^3} \\y &= \frac{(1-\mu)y}{r_1^3} + \frac{\mu y}{r_2^3} \\0 &= \frac{(1-\mu)z}{r_1^3} + \frac{\mu z}{r_2^3}\end{aligned}\tag{11}$$

These three equations have three unknowns that can be solved for the five  $(x, y, z)$  locations. From the third equation,  $z = 0$  is the solution and all five points must lie in the  $x$ - $y$  plane. The two equilateral triangle points can be found when  $r_1 = r_2 = 1$ .

These triangular points are known as  $L_4$  and  $L_5$  and were discovered by Lagrange in

1772. The location of  $L_4$  is  $x = \mu - \frac{1}{2}$ ,  $y = +\frac{\sqrt{3}}{2}$  and the location of  $L_5$  is

$x = \mu - \frac{1}{2}$ ,  $y = -\frac{\sqrt{3}}{2}$ . The other three points were discovered by Euler in 1765,

before Lagrange found the triangular points. Euler found the three collinear points

when  $y = z = 0$ . The three points on the  $x$ -axis can be found from Equation 12.

$$x - \frac{(1-\mu)(x-\mu)}{|x-\mu|^3} - \frac{\mu(x+1-\mu)}{|x+1-\mu|^3} = 0\tag{12}$$

The previous equation can be written as a quintic equation (13) and solved for the three  $x$  values of the collinear points.

$$x^5 + (3 - \mu)x^4 + (3 - 2\mu)x^3 - \mu x^2 - 2\mu x - \mu = 0 \quad (13)$$

Equation (13) never has more than three real roots for  $0 \leq \mu \leq 1$ . It must be solved using numerical techniques to obtain the values for  $x$ . For small values of  $\mu$ ,  $L_1$  and  $L_2$  are on either side of the smaller body and  $L_3$  is on the far side of the larger body. (Szebehely 1967; Wiesel 1989)

### *3.2 Establishing the Stability of the Lagrange Points*

Once the five Lagrange points have been found, the stability of the points should be investigated next to better understand the CR3BP. It is known that if a spacecraft was placed exactly at one of the five points with zero velocity it would stay there and this can be shown mathematically. The question to answer is: “What would happen to a spacecraft if it was placed *near* a Lagrange point?” Linear stability theory can be used to answer this question. Assume that the spacecraft is placed close to a Lagrange point. The velocity at the Lagrange point is zero. The spacecraft’s position is written as

$$x \approx x_e + \delta x, \quad y \approx y_e + \delta y, \quad z \approx \delta z \quad (14)$$

where subscript ‘ $e$ ’ represents the position of the equilibrium or Lagrange point. If the equilibrium point is stable, then  $\delta x$ ,  $\delta y$ , and  $\delta z$  will remain small relative to its initial value. If the equilibrium point is unstable or quasi-stable, then  $\delta x$ ,  $\delta y$ , and  $\delta z$  will increase in magnitude. Equation 14 can be substituted into the equations of motion. This nonlinear system of equations can be linearized to determine the

stability of the five points. The equations are reorganized into state-space form before they are linearized. The next set of equations is the two-dimensional CR3BP in state-space form.

$$\begin{aligned}
\dot{x}_1 &= \dot{x} = x_3 \\
\dot{x}_2 &= \dot{y} = x_4 \\
\dot{x}_3 &= 2\dot{x}_2 + x_1 + \frac{\mu}{r_1^3}(x_1 + 1 - \mu) + \frac{1 - \mu}{r_2^3}(x_1 - \mu) \\
\dot{x}_4 &= -2\dot{x}_1 + x_2 \left( 1 + \frac{\mu}{r_1^3} - \frac{1 - \mu}{r_2^3} \right)
\end{aligned} \tag{15}$$

Next, the state-space representation of the equations of motion needs to be linearized. By definition, the equation below results in a linear set of equations when applied to a nonlinear set of equations.

$$\begin{aligned}
\delta\dot{x} &\approx A\delta x + B\delta u \\
\text{where } A &\equiv \left. \frac{\partial f}{\partial x} \right|_{(\bar{x}, \bar{u})} \quad \text{and} \quad B \equiv \left. \frac{\partial f}{\partial u} \right|_{(\bar{x}, \bar{u})} \\
\text{and } x &= \text{state}, \quad u = \text{input}
\end{aligned} \tag{16}$$

In order to linearize the system of equations, the partial derivatives of the equations of motion are taken with respect to  $x$ ,  $y$ , and  $z$ . Now the eigenvalues and eigenvectors for each Lagrange point can be found from the CR3BP linear set of equations.

If the eigenvalues are all pure imaginary numbers, then the corresponding Lagrange point is stable. If the eigenvalues are real or complex numbers, then the corresponding Lagrange point is unstable. For the two-dimensional ( $x$ - $y$ ) CR3BP, each Lagrange point has four eigenvalues. The collinear Lagrange points,  $L_1$ ,  $L_2$ , and  $L_3$ , each have two real eigenvalues that are equal in magnitude but opposite in sign and two conjugate pure imaginary eigenvalues. The collinear points can be

considered quasi-stable because there are trajectories that will move the third body toward the Lagrange point. The triangular Lagrange points,  $L_4$  and  $L_5$ , each have four purely imaginary eigenvalues.

Two critical values of  $\mu$  can be found from the linear set of equations to explain the stability of the Lagrange points. They are  $\mu_{c1} \approx 0.03852$  and  $\mu_{c2} \approx 0.96148$ . The triangular points,  $L_4$  and  $L_5$ , are stable if  $\mu < \mu_{c1}$  or  $\mu > \mu_{c2}$ . The three collinear points are always unstable for any  $\mu$  ( $0 \leq \mu \leq 1$ ). All of the  $\mu$  values in the Solar System are smaller than 0.03852.

### 3.3 Solving for Zero Velocity Curves

The three-body equations of motion can be written as

$$\begin{aligned}\ddot{x} - 2\dot{y} &= \frac{\partial U}{\partial x} \\ \ddot{y} + 2\dot{x} &= \frac{\partial U}{\partial y} \\ \ddot{z} &= \frac{\partial U}{\partial z}\end{aligned}\tag{17}$$

where  $U$  is the scalar function: 
$$U = \frac{1}{2}(x^2 + y^2) + \frac{1-\mu}{r_1} + \frac{\mu}{r_2}$$

Jacobi's integral derived from Equation 17 can be obtained by multiplying each equation by  $\dot{x}$ ,  $\dot{y}$ , and  $\dot{z}$ , respectively, and then integrating. The resulting equation is

$$\dot{x}^2 + \dot{y}^2 + \dot{z}^2 = 2U - C\tag{18}$$

where  $C$  is a constant of integration. The left-hand side of the equation is the square of the velocity of the third body and can be denoted by  $V^2$  and Equation 18 can then be written as

$$V^2 = 2U - C \quad (19)$$

Equation (19) is Jacobi's integral, also known as the integral of relative energy.

From Jacobi's integral, surfaces of zero velocity can be found for the motion of the third body. When the velocity is zero, Jacobi's integral becomes

$$2U = C$$

$$x^2 + y^2 + \frac{2(1-\mu)}{r_1} + \frac{2\mu}{r_2} = C \quad (20)$$

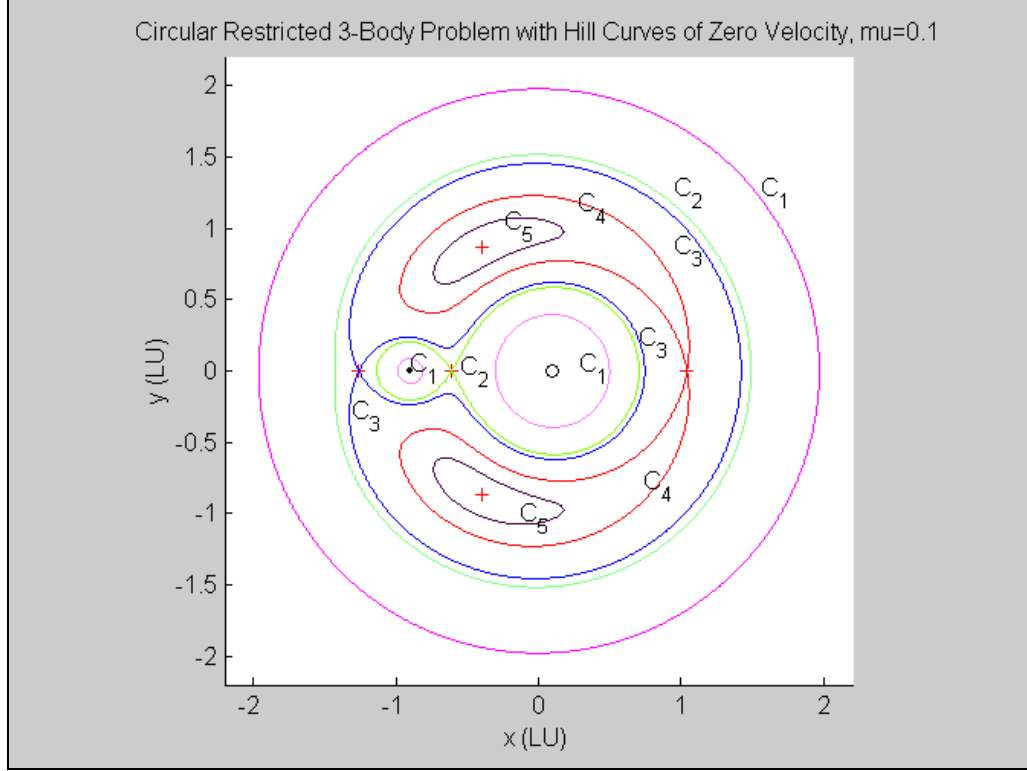
$C$  is known as Jacobi's constant, and it can be determined from initial conditions.

Curves of constant  $C$  values can be plotted to discover more about the motion of the third body. Constant values of  $C$  with zero velocity are known as Hill's curves.

Motion is constrained by  $2U > C$  so that  $V^2$  is positive. If  $2U < C$  then  $V^2$  would be negative, resulting in imaginary velocity values.

The constant  $C$  value curves (also referred to as zero velocity curves) define the boundaries for the third body at a given velocity. The third body is allowed to move to regions with larger  $C$  values, but not to regions with smaller  $C$  values unless a velocity change is applied. Curves for various  $C$  values are shown in Figure 3. Certain specific  $C$  values are important for defining the motion around Lagrange

points and are assigned specific labels from  $C_1$  to  $C_5$ . (The  $C_1$  to  $C_5$  labels do not correspond to increasing values of  $C$ , but instead correspond to decreasing values of  $C$ , such that  $C_1 > C_2 > C_3 > C_4 > C_5$ .) If  $r_1$  and  $r_2$  are very large, then  $C \approx (x^2 + y^2)$ . That approximation results in the equation for a circle and three circles are formed from that approximation, which are labeled  $C_1$  in Figure 3. The three circles include one circle around each of the primary masses and a third circle around the system. If  $C$  is large and either  $r_1$  or  $r_2$  is very small, then the inner curves become separate ovals around the points  $(x, y) = (\mu - 1, 0)$  and  $(x, y) = (\mu, 0)$  and there is a third circle around the system. As the  $C$  value decreases, the inner ovals become larger and the outer circle becomes smaller. The  $C_2$  curve is the specific  $C$  value where the two inner ovals around the two bodies intersect and this intersection point is the Lagrange point,  $L_1$ . As the  $C$  value continues to decrease there is a point where the inner oval of the smaller primary intersects the outer circle of the system and this specific  $C$  value is labeled  $C_3$ . The intersection point of  $C_3$  is defined as Lagrange point,  $L_2$ .  $C_4$  is the specific  $C$  value where the inner oval of the larger body intersects the outer oval of the system and this point is the Lagrange point,  $L_3$ . As the  $C$  value continues to decrease the inner ovals and outer circle continue to converge as seen in  $C_5$ . The points  $L_4$  and  $L_5$  are found where the ovals converge to the outer circle as  $C$  reaches a minimum.



**Figure 3. Curves of constant  $C$  values and zero velocity.**

### 3.4 Generating the Manifolds for the CR3BP

In order to generate a manifold for the CR3BP, the two-dimensional nonlinear equations of motion were numerically integrated. The state-space representation of the equations of motion (Equation 21) was programmed into Matlab.

$$\begin{aligned}
 \dot{x}_1 &= \dot{x} = x_3 \\
 \dot{x}_2 &= \dot{y} = x_4 \\
 \dot{x}_3 &= 2\dot{x}_2 + x_1 + \frac{\mu}{r_1^3}(x_1 + 1 - \mu) + \frac{1 - \mu}{r_2^3}(x_1 - \mu) \\
 \dot{x}_4 &= -2\dot{x}_1 + x_2 \left( 1 + \frac{\mu}{r_1^3} - \frac{1 - \mu}{r_2^3} \right)
 \end{aligned} \tag{21}$$

The set of four, first order, nonlinear equations were numerically integrated using the Runge-Kutta-Fehlberg fourth and fifth order (RKF45) method. The numerical



integration requires an initial position and velocity so that it can generate a trajectory over a specified time interval. An initial velocity of zero is required to find the stable manifold to travel to a Lagrange point. The stable manifold can be propagated forward and backward in time so that a spacecraft is able to travel to and from a Lagrange point on the manifold. The forward time propagation is done using Equation 21. In order to perform backward time propagation, Equation 21 is modified using Equation 22.

$$\begin{aligned}
\frac{dx}{dt} &= f(x) \text{ for forward time propagation} \\
\frac{ds}{dt} &= f(s) = -1 \text{ where } s = -t, \text{ for backward time propagation} \\
\text{so that } \frac{dx}{ds} \frac{ds}{dt} &= \frac{dx}{dt} \\
\text{or } \frac{dx}{ds} &= -\frac{dx}{dt}
\end{aligned} \tag{22}$$

After the modification, the new equations of motion for backward time propagation become Equation 23.

$$\begin{aligned}
\dot{x}_1 &= \dot{x} = x_3 \\
\dot{x}_2 &= \dot{y} = x_4 \\
\dot{x}_3 &= -2\dot{x}_2 + x_1 + \frac{\mu}{r_1^3}(x_1 + 1 - \mu) + \frac{1 - \mu}{r_2^3}(x_1 - \mu) \\
\dot{x}_4 &= 2\dot{x}_1 + x_2 \left( 1 + \frac{\mu}{r_1^3} - \frac{1 - \mu}{r_2^3} \right)
\end{aligned} \tag{23}$$

Equations 21 and 23 were used to generate stable manifolds around each of the five Lagrange points.

### 3.5 Developing the Hamilton-Jacobi-Bellman Optimization Algorithm

In order to examine an optimized solution to the CR3BP, the Hamilton-Jacobi-Bellman (HJB) optimization method was chosen. This method was chosen to calculate optimal trajectories from a location near  $m_1$  or  $m_2$  to a Lagrange point. The optimal control problem was converted into a parameter optimization problem so that it can be solved using nonlinear programming. The nonlinear CR3BP can be described by the following function.

$$\frac{d}{dt}x(t) = f(x, k, t) \quad (24)$$

The state of the system is represented by the state vector  $x(t)$  and the control law is represented by vector  $k(t)$ . Both the state vector and the control are functions of time.

The scalar cost function that is used to describe the optimal control law is

$$J(x, k, t) = \int_t^{t_f} I(x, k, \tau) d\tau + m(x(t_f)) \quad (25)$$

The cost function,  $J$ , integrates the instantaneous cost,  $I$ , over time from  $t$  to  $t_f$  and a final cost,  $m$ , is added to adjust for a difference between the actual state and the desired final state.

The cost function can be written subject to constraints imposed by an optimal control law in order to develop the optimal cost function. Equation 26 is of the optimal cost function, which is now a function of optimal control law,  $k^{opt}$ .

$$\begin{aligned}
J^{opt}(x, t) &= J(x, k^{opt}, t) = \min_k \{J(x, k, t)\} = \\
&\min_k \left\{ \int_t^{t_f} I(x(\tau), k(\tau), \tau) d\tau + m(x(t_f)) \right\} = \\
&\int_t^{t_f} I(x(\tau), k^{opt}(\tau), \tau) d\tau + m(x(t_f)) \\
&\text{subject to } \frac{d}{dt} x(t) = f(x, k, t)
\end{aligned} \tag{26}$$

The optimality principle in dynamic programming allows the optimal control law to be found over a shorter time interval, resulting in Equation 27. (Bellman 1957)

$$J^{opt}(x, t) = \min_k \left\{ \int_t^{t+\Delta t} I(x(\tau), k(\tau), \tau) d\tau + J^{opt}(x(t+\Delta t), t+\Delta t) \right\} \tag{27}$$

From an integral approximation and applying a multi-variable Taylor series expansion, the HJB partial differential equation (PDE) is found and has the following boundary condition in Equation 28. (Dorato et al. 1995)

$$\frac{-d}{dt} J^{opt}(x, t) = \min_k \left\{ I(x, k, t) + \left[ \frac{d}{dx} J^{opt}(x, t) \right]^T f(x, k, t) \right\} \tag{28}$$

$$\text{where } J^{opt}(x(t_f), t_f) = m(x(t_f))$$

An analytical solution to the HJB PDE is only possible for special cases, so when applying the HJB PDE to the CR3BP, a numerical approach must be used to find a solution to the optimal control problem. The chosen numerical approach approximates the continuous-time optimal control problem as a finite-parameter constrained optimization problem that can be solved by nonlinear programming. With this approach, a trajectory in the CR3BP is propagated using a finite number of nodes that contain dynamic information including the applied control and continuous-time position and velocity. The trajectory is divided into time intervals with the final

time for each interval named a node. At each node the state (position and velocity) and controller are discretized, so that there are a finite number of equality constraints placed on the dynamics. The controller variables were chosen to be applied as an impulsive change in velocity,  $\Delta V$ , to the body at each node. Once the state and controller are discretized, the cost function (29) can be written as a summation over the number of nodes (n-1).

$$J(x, k, t) = \sum_{i=1}^{n-1} I(x_i, k_i, \tau_i) \Delta \tau + m(x(t_f)) \quad (29)$$

The control applied over each segment was chosen to remain constant until the next node. At each node, a new control can be applied, resulting in a piecewise-constant control input over the trajectory. The system dynamics used to propagate the trajectory were performed using the Runge-Kutta-Fehlberg (RKF45) numerical integration method. In order to satisfy the system dynamics of the CR3BP for a given segment, the variables at either end of the segment were modified by the nonlinear programming solver, so that the constraints are met within a user-specified tolerance. (Epperson 2002; Stoer and Bulirsch 1993; von Stryk and Schlemmer 1994)

#### Structure of Nonlinear Programming Algorithm (Bhatti 1998)

Initial guess is provided for each node vector as a starting solution.

The first derivative of the node vectors is calculated and the second derivative of the cost is approximated with respect to the node vectors.

The new state of the next node is determined based on reducing the cost from the information in step 2.

The new solution is found from the new state of each node.

Go to step 2 and repeat until the cost stops decreasing.

Once a new minimum cost has been found, it is compared to the original cost. If the new cost is greater than the allowable tolerance applied to the original cost, then a converged solution to the constrained cost problem has not yet been found. If the solution has not yet converged, then the most recent solution is used as a new guess for step 1.

An impulsive velocity ( $\Delta V$ ) model was applied at each node in order to adjust the state vector for the following segment. The impulsive velocity cost is a finite sum of the  $\Delta V$ 's. Equation (30) is the cost that needs to be minimized over the entire trajectory.

$$Cost_{vehicle} = \sum_{node=1}^n \Delta V_{node} \quad (30)$$

Now the HJB optimized solution to the CR3BP has been found through programming algorithms. The optimization code that was written can now be run for the CR3BP with varying initial and final states. The following parameters can be varied for optimization of different cases: gravitational parameter ( $\mu$ ), starting position and velocity, ending position and velocity, and time of trajectory.

## 4. Results

### *4.1 Manifold Solution of the CR3BP*

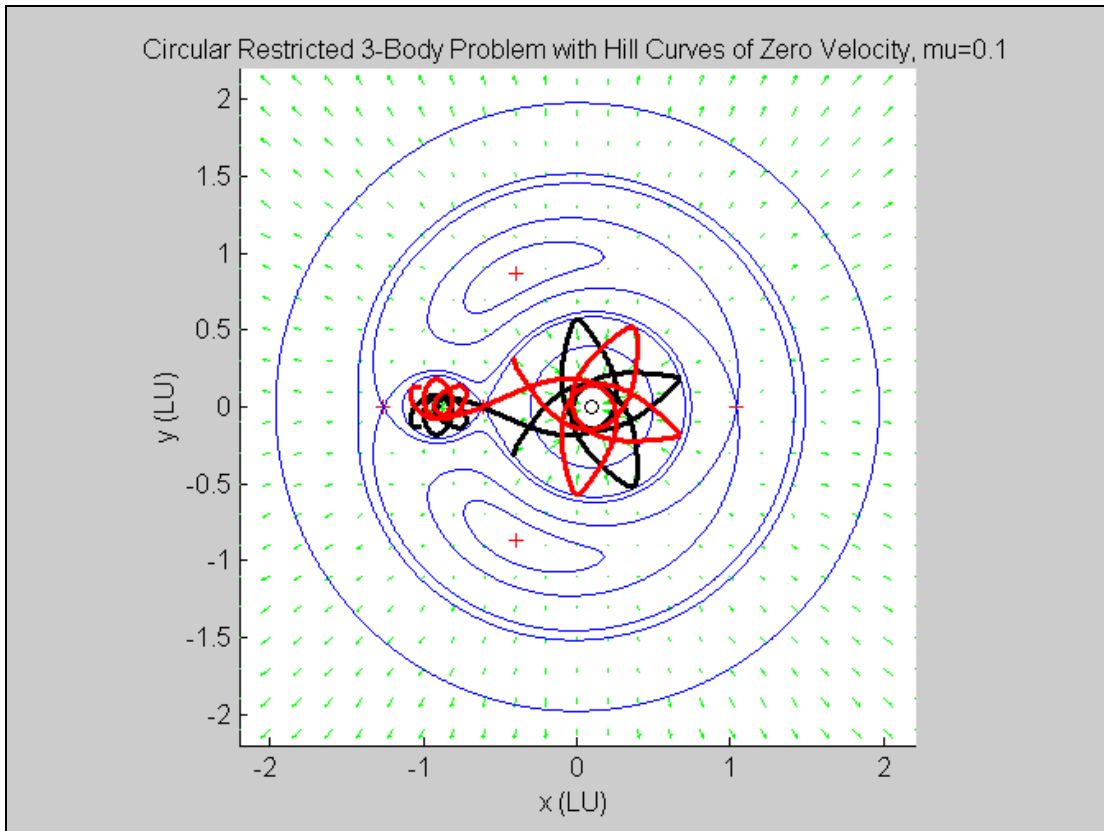
Matlab was used in order to program everything that was needed to develop the three-body manifolds. First, the five Lagrange points were found from the CR3BP equations of motion as described in the methods section. Next, the equations of motion were linearized in order to find the eigenvalues and eigenvectors to better understand the system. The eigenvectors were plotted to better understand the motion of the third body and obtain approximate locations of the two-dimensional manifolds. The eigenvectors show direction of motion around each Lagrange point. The potential,  $U$ , was found for the CR3BP and Jacobi's constant of integration,  $C$ , was calculated. Hill's curves were plotted with zero velocity and constant values of  $C$ . The curves were used to define areas of restricted motion for the third body.

Motion of the third body was found using numerical integration of the CR3BP nonlinear equations of motion as described in the methods section. An initial position and velocity was assigned to the third body and then propagated over a specified time interval. The method of numerical integration used for the trajectory propagation was Runge-Kutta-Fehlberg (RKF45) with equations from the 4<sup>th</sup> and 5<sup>th</sup> order methods. Depending on the initial velocity given to the third body (no additional velocity applied), its motion was restricted to clearly defined areas that can be shown once the potential,  $U$ , and Jacobi's constant,  $C$ , are calculated.

Dynamic systems theory was used to propagate the third body's motion in order to find the stable manifold in the two-dimensional  $x$ - $y$  plane. No additional acceleration or velocity is applied to the third body so that the stable manifold can be found only from the dynamics. In order to have the third body travel to a Lagrange point, using only dynamics, it must be on a stable manifold that will allow it to move toward that point. The third body needs to approach a trajectory defined by the eigenvectors about a Lagrange point, so that it can travel to one of the points.

The stable manifolds around each Lagrange point were generated using only "natural" dynamics. For each point, four locations were chosen to initiate dynamic propagation in order to find the stable manifold. If the third body is propagated forward in time from a point on the stable manifold it will move toward a Lagrange point, but if the third body is propagated from an unstable manifold it will move away from a Lagrange point. Each of the four locations were displaced from the Lagrange point by  $\pm 0.0001$  in the  $x$  and  $y$  directions, such that each of the four quadrants around the Lagrange point were represented by one of the initial locations. Each of the collinear Lagrange points ( $L_1, L_2, L_3$ ) have stable directions designated by the eigenvectors where the third body can move toward the Lagrange point. The stable pathways were found by propagating the dynamics backward in time from a position near the Lagrange point. For the collinear Lagrange points, the unstable pathways were found by forward time propagation of the third body from positions near the Lagrange point. The triangular Lagrange points ( $L_4, L_5$ ), which are stable, allow the third body to move toward the Lagrange point from any direction.

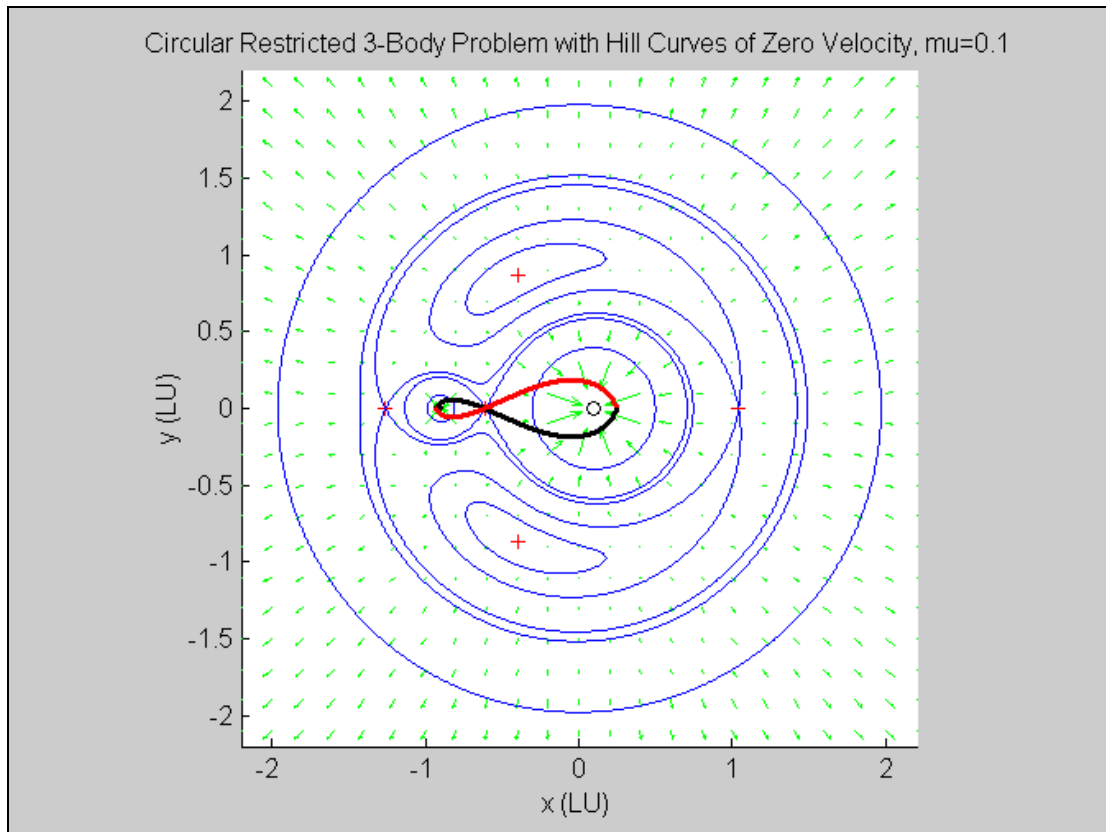
Figure 4 shows the third body's motion propagated from positions near  $L_1$ . The dynamics were propagated with zero initial velocity. The black curves are the unstable manifolds and were found from forward time propagation. The red curves are the stable manifolds, which were found by propagating the dynamics backward in time from points near  $L_1$ . The stable and unstable manifolds around  $L_1$  are constrained by the  $C_2$  zero velocity curve. If the third body starts on the side closer to  $m_1$  (right side of  $L_1$ ), then the third body's motion remains trapped around  $m_1$ . If the third body starts on the side closer to  $m_2$  (left side of  $L_1$ ), then the third body's motion remains trapped around  $m_2$ .



**Figure 4. The  $L_1$  manifold propagated over a long period of time.**

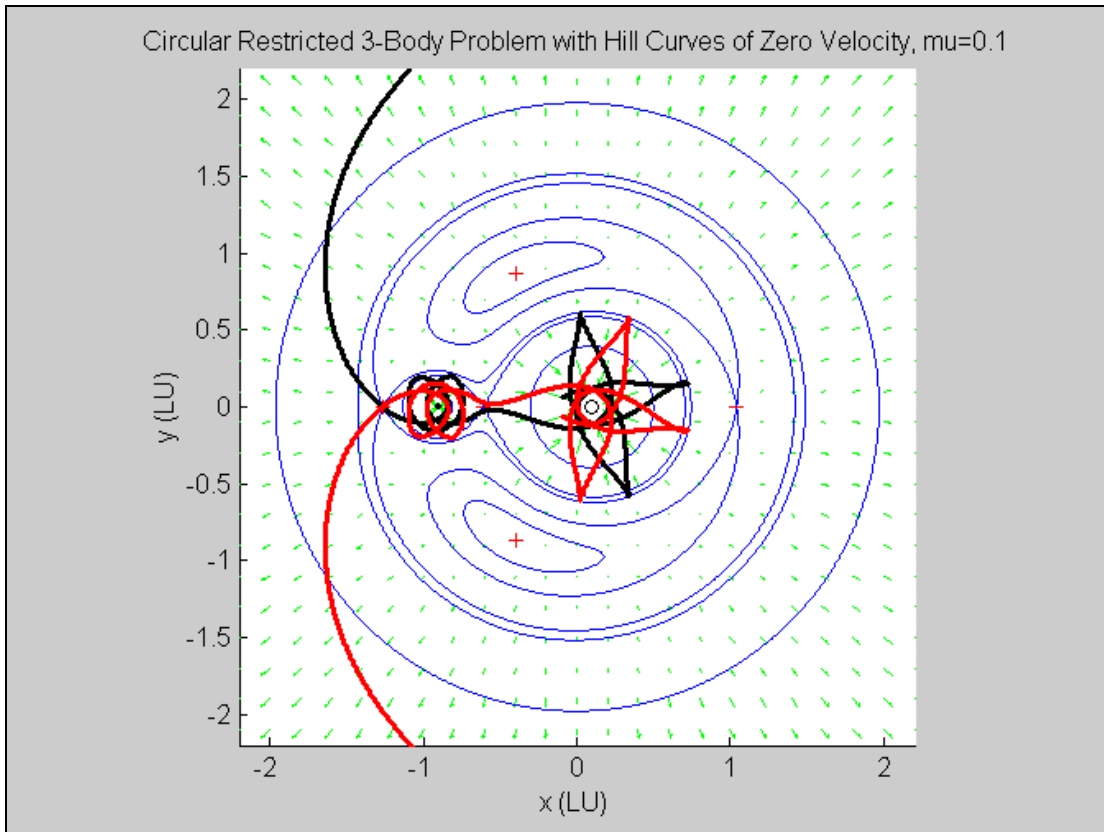


Figure 5 shows the  $L_I$  manifold propagated over a shorter time than the manifold in Figure 4. This portion of the manifold is used to compare to optimized control solutions in section 4.3.



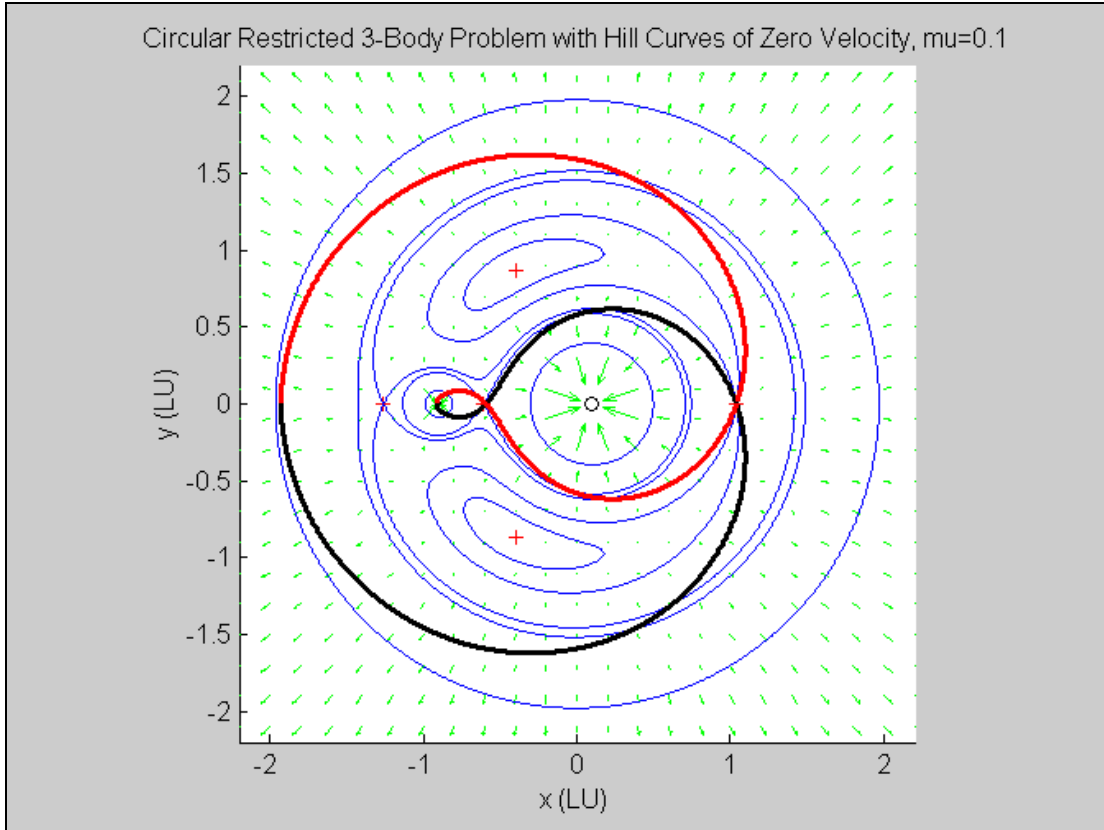
**Figure 5. The  $L_I$  manifold propagated over a short period of time.**

Figure 6 shows the third body's motion propagated from positions near  $L_2$ . The black curves are the unstable manifolds and were found from forward time propagation. The red curves are the stable manifolds, which were found by propagating the dynamics backward in time from points near  $L_2$ . The stable and unstable manifolds around  $L_2$  are constrained by the  $C_3$  zero velocity curve. If the third body starts on the side closer to  $m_1$  and  $m_2$  (right side of  $L_2$ ), then the third body initially orbits  $m_2$  and then moves into orbit about  $m_1$ . If the third body starts on the side farther away from  $m_1$  and  $m_2$  (left side of  $L_2$ ), then the third body moves continually away from the system.



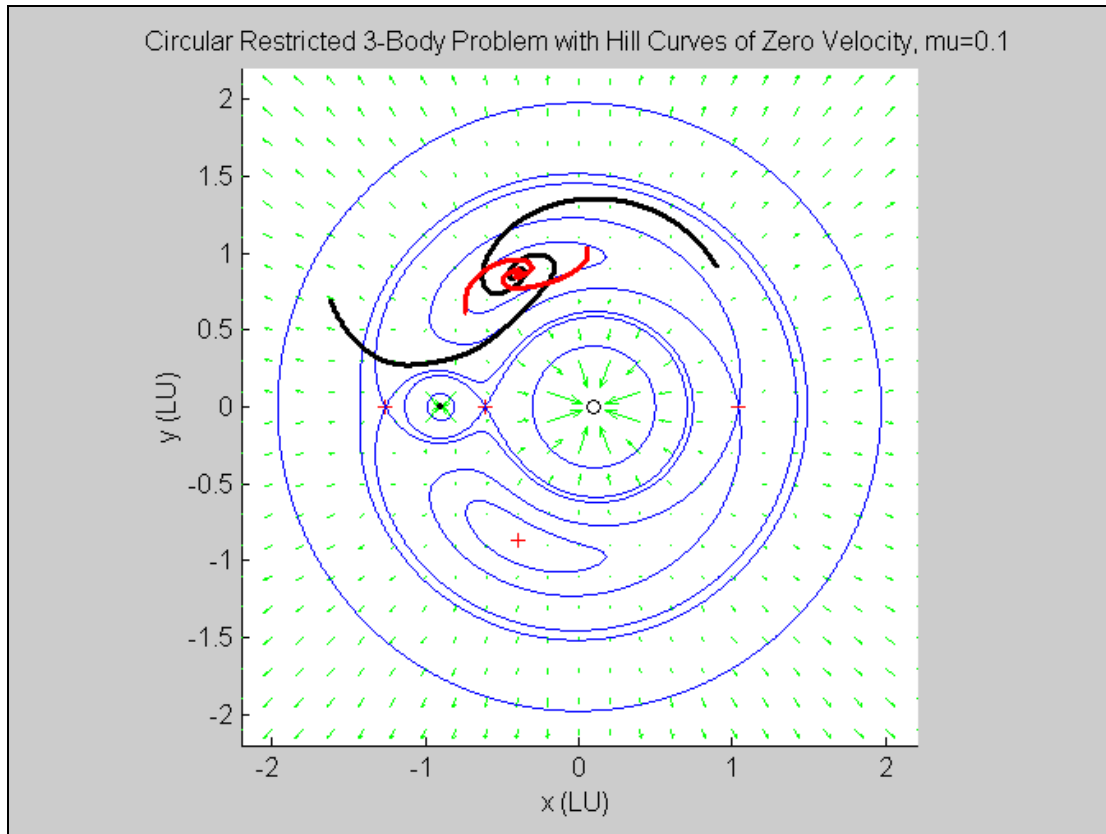
**Figure 6. The  $L_2$  manifold.**

Figure 7 shows the third body's motion propagated from positions near  $L_3$ . The black curves are the unstable manifolds and were found from forward time propagation. The red curves are the stable manifolds, which were found by propagating the dynamics backward in time from points near  $L_3$ . The stable and unstable manifolds around  $L_3$  are constrained to remain outside of the  $C_4$  zero velocity curve. If the third body starts on the side closer to  $m_1$  and  $m_2$  (left side of  $L_3$ ), then the third body initially travels past  $m_1$  and then orbits  $m_2$  for a time. Then third body moves into orbit about  $m_1$ . If the third body starts on the side farther away from  $m_1$  and  $m_2$  (right side of  $L_3$ ), then the third body moves continually away from the system.



**Figure 7. The  $L_3$  manifold.**

Figure 8 shows the third body's motion propagated from positions near  $L_4$ . The black curves were found from forward time propagation and the red curves were found by propagating the dynamics backward in time from points near  $L_4$ . If the third body starts at exactly the  $L_4$  or  $L_5$  location, it will remain there because those points are stable. If the third body starts near  $L_4$  or  $L_5$ , then the motion is unconstrained. The third body is initially influenced by the system but continues on an erratic trajectory.



**Figure 8. The  $L_4$  manifold.**

This section has shown the manifolds around the Lagrange points of the CR3BP. The stable and unstable manifolds agree with the motion described by the eigenvectors.

Zero velocity curves were also used to verify the dynamic motion of the third body.

The manifolds generated have the potential to be used to travel to and from Lagrange points more optimally. The next section uses optimal control theory to find trajectories that will be compared to the manifolds.

#### *4.2 HJB Optimized Solution of the CR3BP*

An optimized solution algorithm was created using the HJB equation applied to the CR3BP. The optimization code finds an optimal trajectory for the third body with a given initial and final state over a period of time. The total cost of the trajectory is calculated from the impulsive velocities applied at nodes (equally spaced time segments) along the trajectory.

Optimized solutions were found for trajectories along the manifold through  $L_1$ . The  $L_1$  manifold was chosen for optimized solutions because this point has been used by numerous satellites in the Sun-Earth system. The  $L_1$  point in the Earth-Moon system will probably be used by spacecraft in the near future because of the NASA goals to return to the moon and travel further out in space and  $L_1$  provides many advantages for doing that. The  $L_1$  manifold is also an interesting manifold where travel from either  $m_1$  or  $m_2$  to the point can be studied.

The HJB optimization algorithm was used with the  $L_1$  manifold for  $\mu = 0.5$  and  $0.1$  and all of the cases were run using 10 nodes. The number of nodes affects the trajectory that the third body takes to achieve its desired final state and the overall cost of the trajectory will vary depending on the number of nodes. The nodes are placed at the end of evenly spaced time intervals throughout the trajectory and are the locations where the impulsive  $\Delta V$ 's are applied. The number of nodes chosen for these optimization cases was ten because it is sufficient to calculate a relatively smooth trajectory but does not require a large amount of time to run each case.

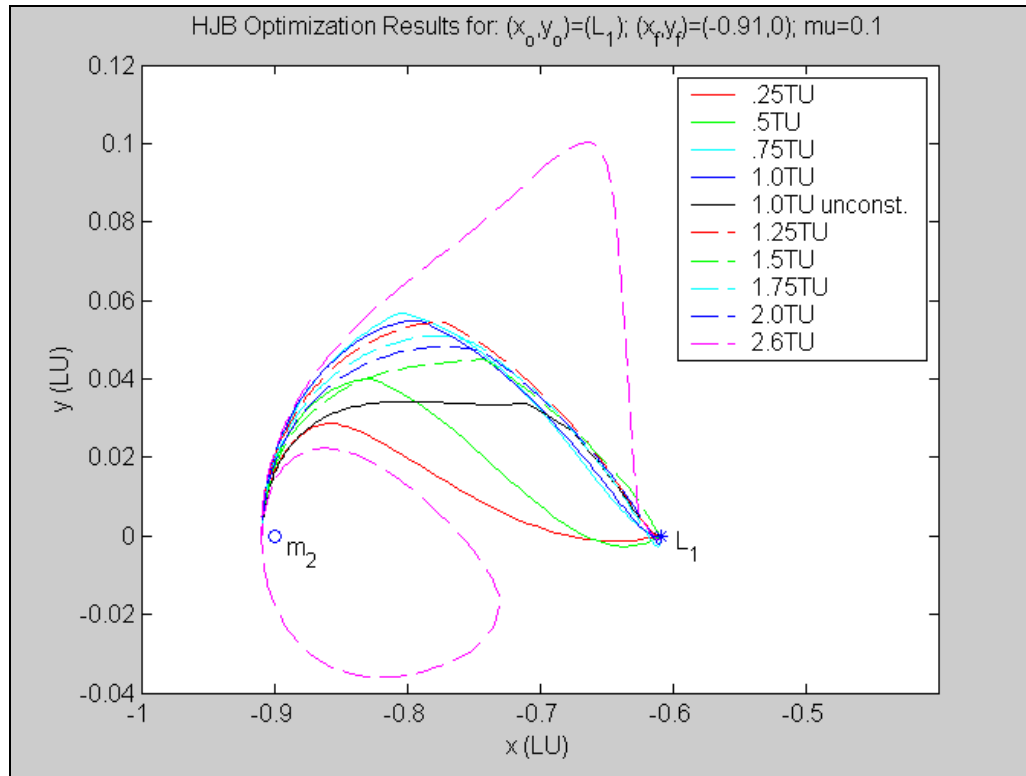
Increasing the number of nodes results in an optimization problem with a greater number of parameters. Such a case then requires more time to compute since the optimization algorithm has a greater number of parameters to vary while minimizing the cost.

Initial optimized solutions were found for  $\mu = 0.5$  to verify the dynamics when the two primary masses are equal. The dynamics were verified by comparing trajectories to  $m_1$  and  $m_2$  that should be the same. More solutions were found for  $\mu = 0.1$  because this is a more interesting case where  $m_1$  is much larger than  $m_2$  and the solutions are more comparable to real life scenarios of the Sun-Earth-Moon system. This section presents the results found using the HJB optimization algorithm and the next results section provides more analysis of the optimized solutions by comparing them to manifolds.

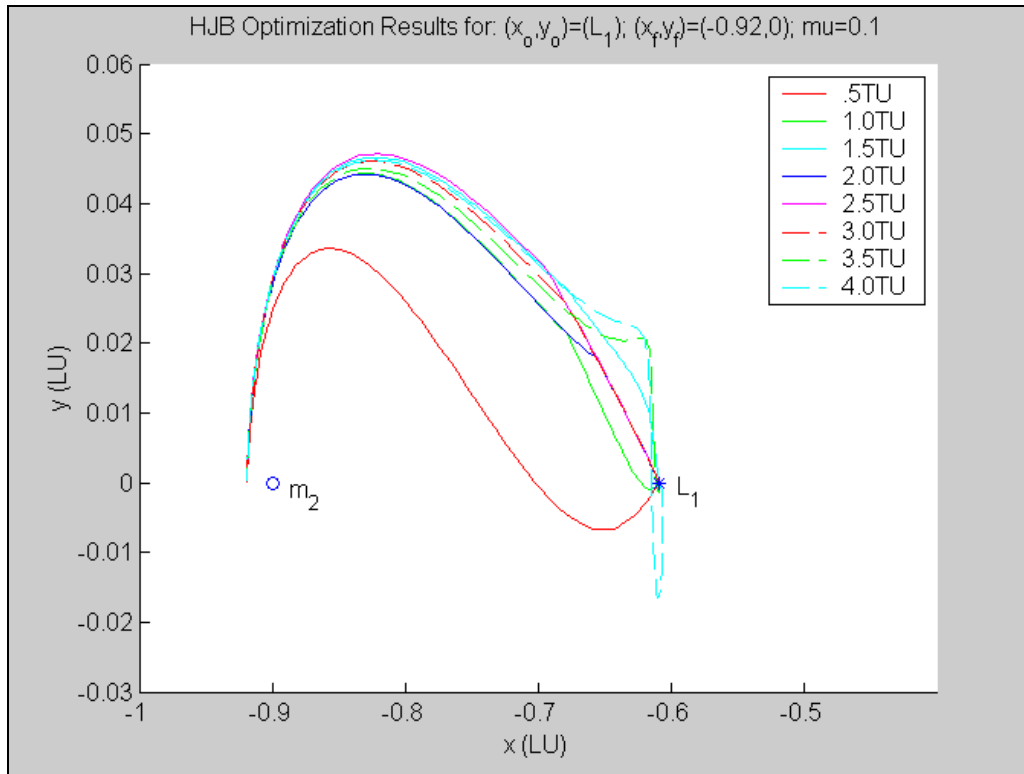
The following three figures show the results from varying cases when  $\mu = 0.1$ . Figure 9 shows optimized solutions for the starting state of  $(x, y, \dot{x}, \dot{y}) = (-0.60903511002320, 0, 0, 0)$  which is the position of  $L_1$ , the point between  $m_1$  and  $m_2$ . The final state was chosen to be  $(x, y, \dot{x}, \dot{y}) = (-0.91, 0, -3.16, 0)$ , where the final position was to the left of  $m_2$  but inside of the manifold through  $L_1$  and the final velocity was set to be a counterclockwise circular velocity around  $m_2$ . Equation 31 was used to calculate the circular velocity, where  $G$  is the gravitational constant for mass,  $m$ , and the orbit has radius,  $r$ .

$$v_{circ} = \sqrt{\frac{Gm}{r}} \quad (31)$$

These solutions, with the previously mentioned initial and final states, were found for a variety of times, from 0.25TU to 2.60TU.



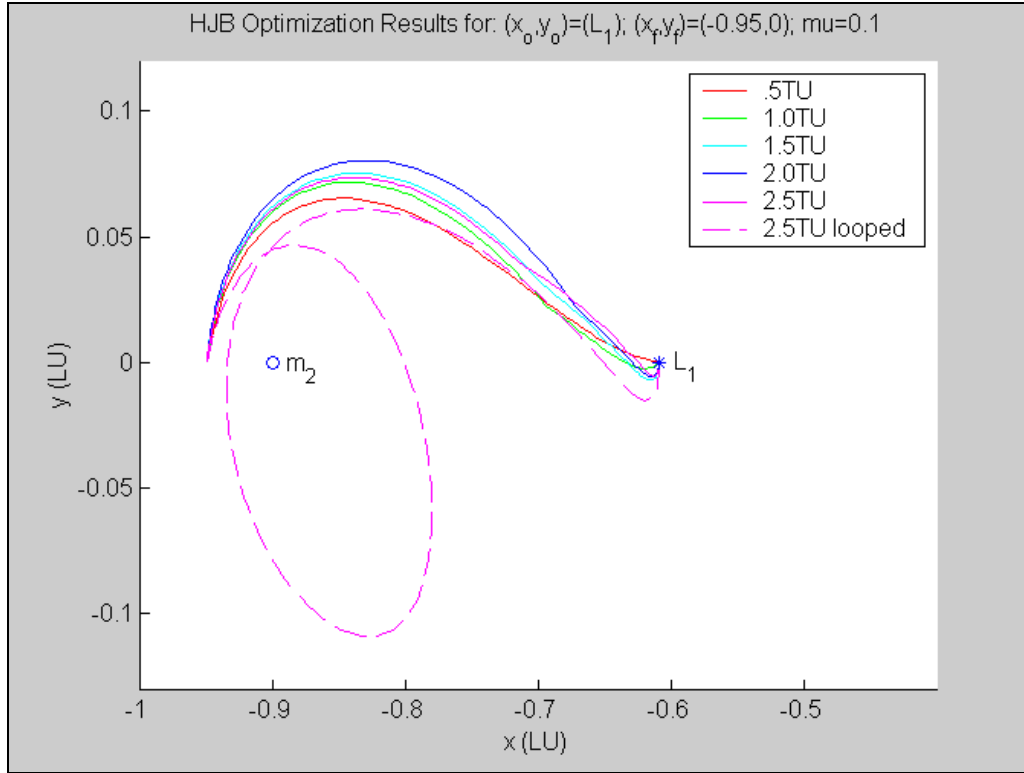
**Figure 9. HJB optimization results from  $L_1$  to  $(-0.91, 0)$ .**



**Figure 10. HJB optimization results from  $L_1$  to  $(-0.92, 0)$ .**

Figure 10 shows optimized solutions for the same initial state as Figure 9, which is  $(x, y, \dot{x}, \dot{y}) = (-0.60903511002320, 0, 0, 0)$ . The final state was set to  $(x, y, \dot{x}, \dot{y}) = (-0.92, 0, \text{unconstrained}, \text{unconstrained})$ , where the final position was left of  $m_2$  but inside of the manifold through  $L_1$ . The final velocity was set to be unconstrained so that the end velocity can vary and the optimization algorithm is able to solve for the end velocity that provides the lowest cost for the entire trajectory. These solutions were found for times from 0.5TU to 4.0TU.





**Figure 11. HJB optimization results from  $L_1$  to  $(-0.95, 0)$ .**

Figure 11 shows optimized solutions for the same starting state as Figures 9 and 10, which is  $(x, y, \dot{x}, \dot{y}) = (-0.60903511002320, 0, 0, 0)$ . The final state was set to  $(x, y, \dot{x}, \dot{y}) = (-0.95, 0, \text{unconstrained}, \text{unconstrained})$ , where the final position was to the left of  $m_2$  and outside of the manifold through  $L_1$  and the final velocity was set to be unconstrained so that the end velocity can vary. These solutions were found for times from 0.5TU to 2.5TU.

#### 4.3 Comparison of Manifold and HJB Optimized Solutions About the $L_1$ Point

In order to compare the manifold approach and the HJB optimized approach for trajectories in the CR3BP, an area of interest had to be chosen. Travel around each of the Lagrange points provides for interesting results from a dynamics standpoint.

There are three-dimensional manifolds that can be examined in the CR3BP but for this project, two-dimensional manifolds were examined to better understand manifold behavior in the CR3BP. One interesting two-dimensional manifold in the CR3BP is the manifold that goes through  $L_1$ , the Lagrange point between the two primary masses, and forms a figure-eight curve around  $m_1$  and  $m_2$ . This manifold was found by propagating the dynamics at points close to  $L_1$  with zero velocity. The HJB optimization was run for  $\mu = 0.1$  from  $L_1$  with zero initial velocity and was allowed to have an unconstrained end position and velocity. The optimized solution was compared to the manifold generated in Matlab and the two curves overlap throughout the entire trajectory, verifying the manifold to be the “natural, unpowered” dynamic approach. This case and all future optimized cases were run for ten nodes on each trajectory.

There are many questions that need to be answered in regard to manifolds and the CR3BP. How do trajectories near the manifold behave? If a small spacecraft (third body) needs to travel from the vicinity of one of the primary masses, what pathway will it take to get to  $L_1$ ? Will it take the same pathway as the manifold or take another pathway? If it approaches the manifold, when along its trajectory will it approach or cross the manifold? These questions can be answered by comparing manifolds to optimized trajectories in the CR3BP.

In order to answer these questions, several cases were run using the HJB optimization algorithm for the CR3BP. All of the cases were run using ten nodes where the  $\Delta V$

was applied. First, an optimization was performed for an initial position at  $L_1$  with zero velocity and a final position of  $(x, y) = (-0.91, 0)$  with an unconstrained velocity. The unconstrained velocity case is shown in Figure 12. The optimized solution followed the pathway of the manifold for part of its trajectory and then the optimizer applied a  $\Delta V$  that caused the body to move toward its final state. This case was performed with a total cost of 0.1887LU/TU. This case needed to be performed before cases with constrained end velocities so that the optimizer had an approximate initial guess at each node.

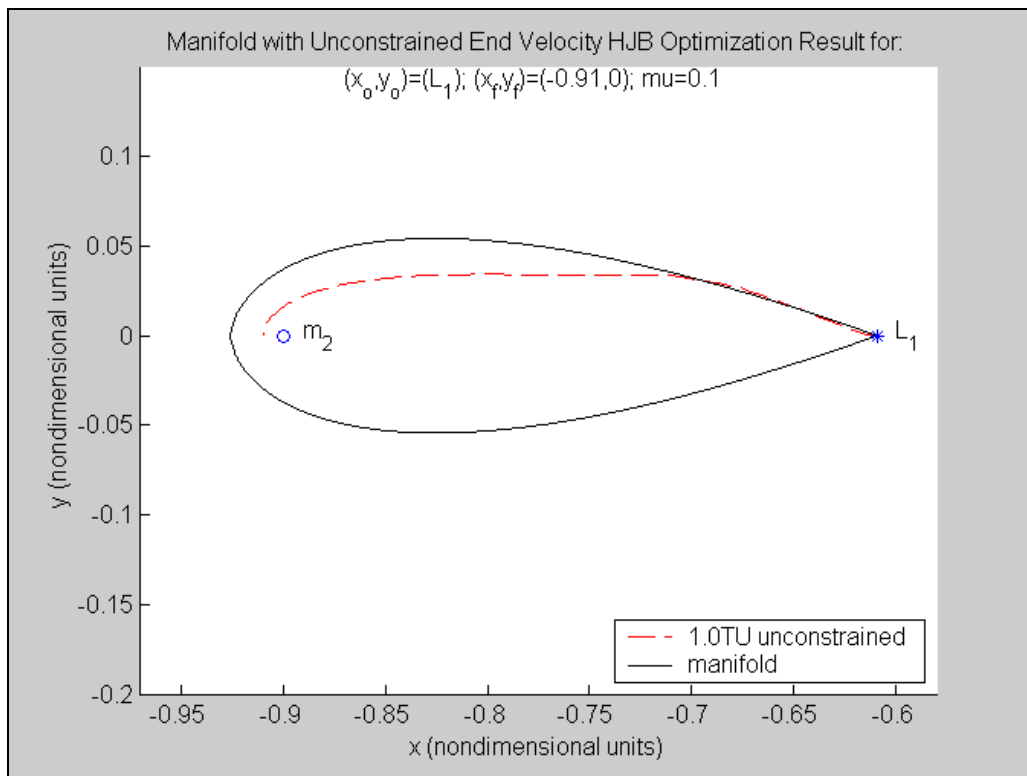
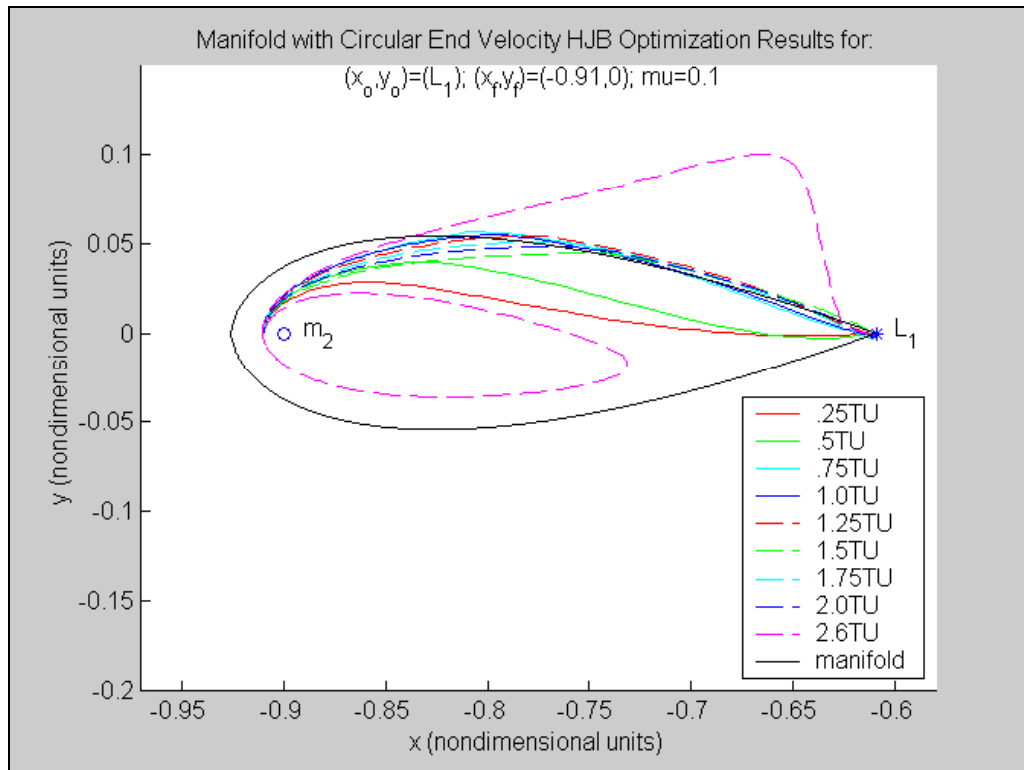


Figure 12. Comparison of the manifold to HJB optimization results from  $L_1$  to  $(-0.91, 0)$  with an unconstrained end velocity.

After approximate node velocities were obtained from the unconstrained case, the first set of cases were chosen to start at  $L_1$  with zero velocity and they were optimized over a trajectory to a point near  $m_2$  that is inside of the  $L_1$  manifold, specifically at the point  $(x, y) = (-0.91, 0)$ . The final state was constrained to have a counterclockwise circular velocity of approximately  $(\dot{x}, \dot{y}) = (-3.16, 0)$  around  $m_2$ . These cases were run in order to determine how the optimized trajectories would relate to the manifold. The trajectories from the initial to the final states mentioned above were optimized for times of 0.25TU to 2.6TU. The optimized solutions and the manifold are shown in Figure 13. The cost analysis and distance following the manifold found from these cases are summarized in Table 1.



**Figure 13. Comparison of the manifold to HJB optimization results from  $L_1$  to  $(-0.91, 0)$ .**

<b>Integration Time (TU)</b>	<b>Distance Following Manifold Within 0.01LU (LU)</b>	<b>Cost (LU/TU)</b>
1.0	0.14	0.1887*
0.25	0.015	2.565
0.5	0.015	1.756
0.75	0.26	1.582
1.0	0.26	1.545
1.25	0.245	1.481
1.5	0.18	1.362
1.75	0.24	1.505
2.0	0.235	1.489
2.6	0.02	1.619

**Table 1. Results for Case 1:  $x_0 = (-.609, 0, 0, 0)$ ;  $x_f = (-.91, 0, -3.16, 0)$   
\*unconstrained end velocity**

The 0.25TU case forced the trajectory to leave the manifold as soon as it began because the time was too short for the body to take a path of less  $\Delta V$  required. The cost required for this trajectory was 2.565LU/TU. This cost was much higher than the unconstrained case because it was forced to find a solution in less time and it was forced to have a specific final velocity. The 0.5TU case also left the manifold immediately because of its small time allowed to reach its final state. The largest time case that was run for these initial and final states was for 2.6TU. This time was much too large and the trajectory strays from the manifold right away. This case also required a large  $\Delta V$  relative to the unconstrained case. The 2.6TU case had a cost of 1.619LU/TU. The cases for 0.75TU to 2.0TU all started out by following the manifold pathway and then leave the manifold at various points because they are

forced to achieve their final state. The case with the lowest cost for the circular end velocity was for 1.5TU and the cost was 1.362LU/TU. The distance that each trajectory followed the manifold within 0.01LU was measured using plot comparisons of each trajectory to the manifold. The distance of 0.01LU was arbitrarily chosen as the tolerance for being considered “on the manifold” in order to make comparisons between the optimized trajectories. From Figure 13, the cases that followed the manifold, within 0.01LU, for the longest distance were for times of 0.75TU and 1.0TU. There does not seem to be a direct correlation between the cost and the time spent on the manifold for the 0.75TU to 2.0TU cases. The most likely reason for miscorrelation is that these cases were run with ten nodes and the nodes for each trajectory are in different locations because the nodes are evenly spaced over time, not distance. The definition of the nodes results in different trajectories leaving the manifold at different locations depending on where the node occurs and the velocity change at that node. This could be investigated further if this set of cases were run with more nodes, which may produce a case where the lowest cost is found to have spent the greatest distance on the manifold.

The next set of cases that were run with the HJB optimization algorithm had an initial state at the  $L_1$  position and zero velocity and the optimized trajectories shown with the manifold in Figure 14. The final position is still inside of the  $L_1$  manifold around  $m_2$ , but the position is further away from  $m_2$  than the previous set of cases in Figure 13. The final state had a position of  $(x, y) = (-0.92, 0)$  and an unconstrained velocity. The

trajectories were optimized for times from 0.5TU to 4.0TU. The cost analysis and distance following the manifold found from these cases are summarized in Table 2.

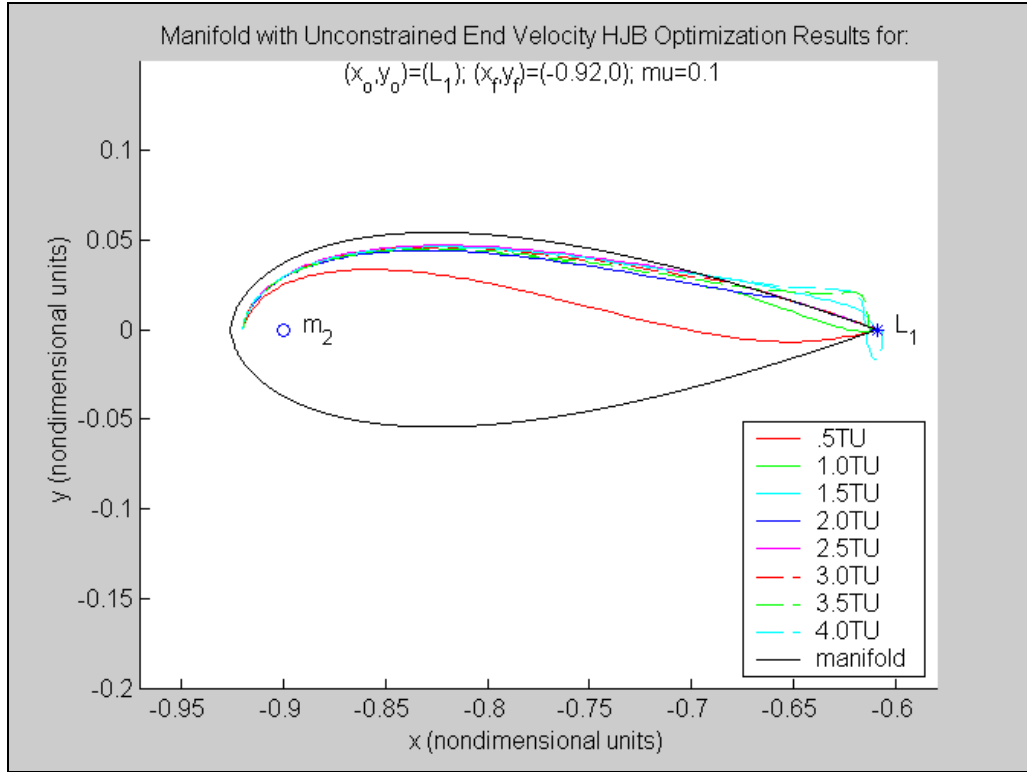


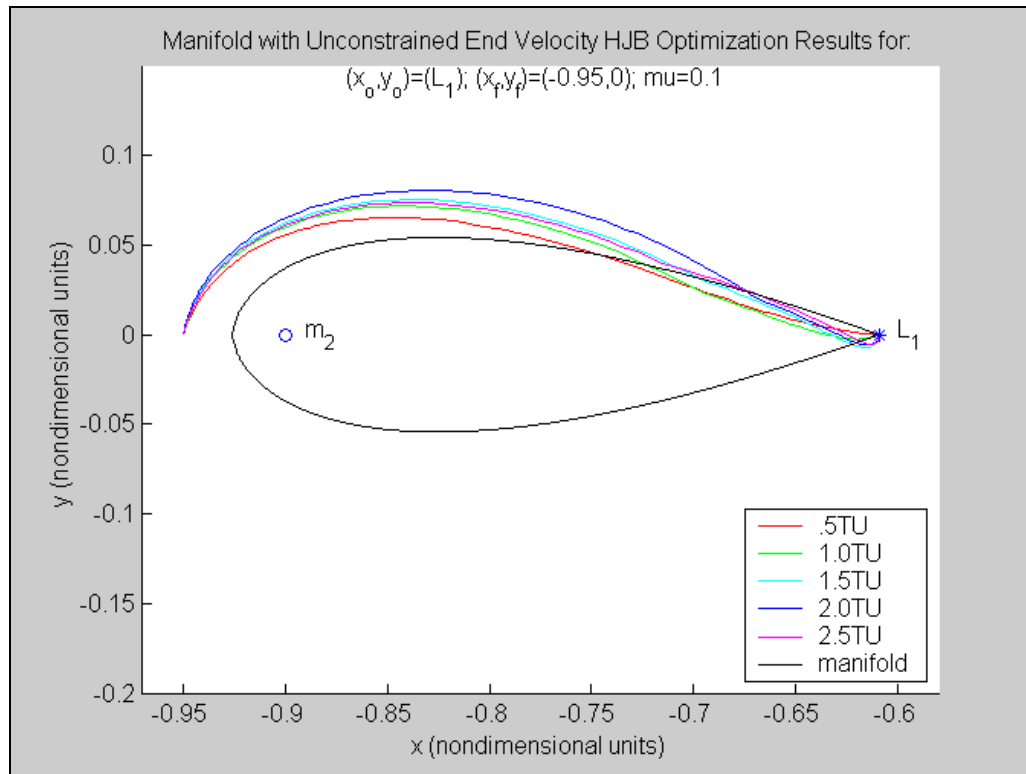
Figure 14. Comparison of the manifold to HJB optimization results from  $L_1$  to  $(-0.92, 0)$ .

Integration Time (TU)	Distance Following Manifold Within 0.01LU (LU)	Cost (LU/TU)
0.5	0.02	0.2836
1.0	Entire Distance	0.1000
1.5	Entire Distance	0.04499
2.0	Entire Distance	0.04103
2.5	Entire Distance	0.03982
3.0	Entire Distance	0.03862
3.5	<0.01	0.03838
4.0	<0.01	0.03831

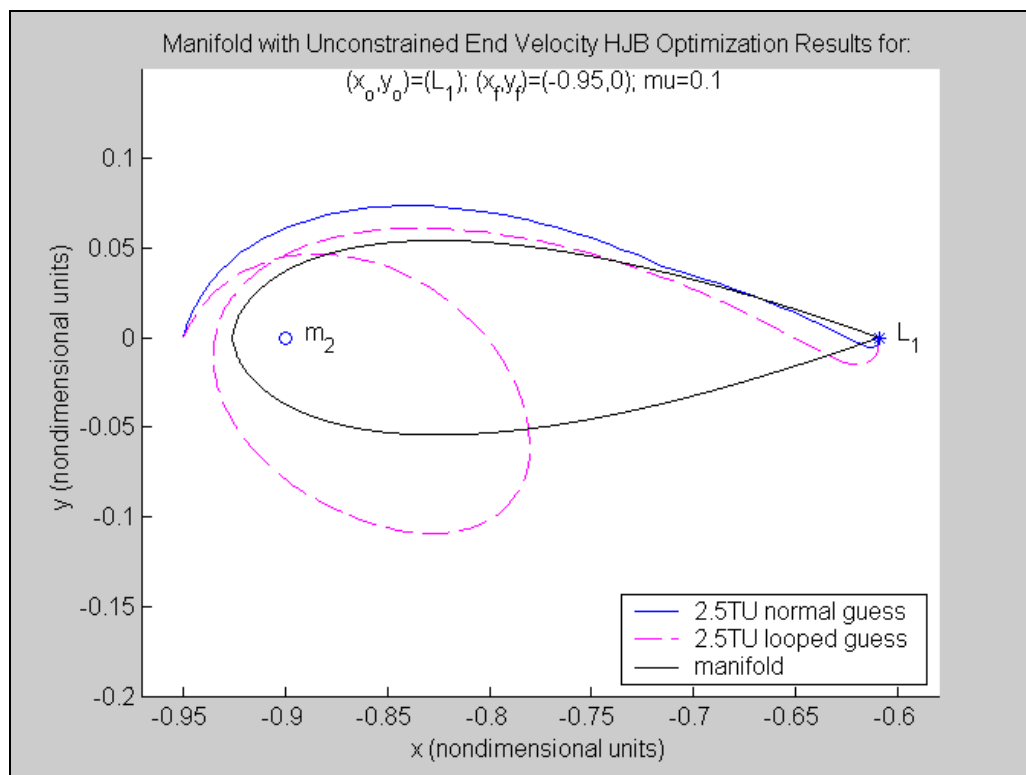
Table 2. Results for Case 2:  $x_0 = (-0.609, 0, 0, 0)$ ;  $x_f = (-0.92, 0, \text{unconstrained}, \text{unconstrained})$

From Figure 14, the 0.5TU case was not allowed enough time for the trajectory to follow the manifold at all. It had a cost of 0.2836LU/TU, which was the largest cost found for this set of cases. The time cases of 1.0TU and 1.5TU are initially farther from the manifold than the cases of 2.0TU, 2.5TU, and 3.0TU, but all five of these cases remain within the 0.01LU tolerance of the manifold for the entire trajectory. The 3.5TU and 4.0TU cases did not follow the manifold pathway within the 0.01LU tolerance initially, but fairly early their optimized trajectories returned to the manifold and remained within the tolerance until the final state. For this set of cases in Figure 14, unlike the cases in Figure 13, the cost varied as expected with time so that more time allowance was associated with less cost. The lowest cost was found for the 4.0TU case and was 0.03831LU/TU, even though this trajectory left the manifold initially.





**Figure 15. Comparison of the manifold to HJB optimization results from  $L_1$  to  $(-0.95, 0)$ .**



**Figure 16. Comparison of the normal and looped cases of HJB optimization results from 2.5TU.**

The third set of cases optimized using the HJB programming algorithm was performed with a final position that is outside of the  $L_1$  manifold around  $m_2$ . These cases are shown in Figures 15 and 16. All of the cases were run with an initial state at  $L_1$  and zero initial velocity. The final state was set to  $(x, y) = (-0.95, 0)$  and an unconstrained velocity. The trajectories were optimized for the times of 0.5TU to 2.5TU. The cost analysis and distance following the manifold found from these cases are summarized in Table 3. The lowest cost was found for 2.0TU and the total cost was 0.1066LU/TU. The 0.5TU trajectory followed the manifold for the longest distance, but also had the highest cost from this set of cases. This is probably because the final position of the trajectories is outside of the manifold and a more direct path to the endpoint will follow the manifold for a longer distance. The cost decreased with increasing time for the 0.5TU to 2.0TU trajectories, but the cost increased for the 2.5TU case. In the 2.5TU case, it appears as if too much time is allowed for the trajectory to reach its endpoint, causing the cost to increase due to the increased distance traveled relative to the time allowed.

<b>Integration Time (TU)</b>	<b>Distance Following Manifold Within 0.01LU (LU)</b>	<b>Cost (LU/TU)</b>
0.5	0.24	0.3779
1.0	0.17	0.1468
1.5	0.13	0.1077
2.0	0.11	0.1066
2.5 normal guess	0.14	0.1130
2.5 looped guess	<0.01	0.1208

**Table 3. Results for Case 3:  $x_0 = (-.609, 0, 0, 0)$ ;  $x_f = (-.95, 0, \text{unconstrained}, \text{unconstrained})$**

In Figure 16, two cases were compared for 2.5TU and have the same initial and final states as in Figure 15. These two optimized trajectories were run with different initial guesses of their expected pathways. The “normal guess” was given a more direct initial approximation that would take it on a pathway similar to the manifold and to its final state. The “looped guess” was given an initial trajectory approximation that would loop around  $m_2$  before moving to its designated final state. The direct route case had a cost of 0.1130LU/TU, while the looped case had a greater cost of 0.1208LU/TU.

## 5. Conclusions

Matlab programming was used to develop manifolds for the CR3BP. The five Lagrange points were found and the equations of motion were linearized to find the eigenvectors and eigenvalues. Hill’s zero velocity curves were plotted for constant values of  $C$ . The curves were used to define areas of restricted motion for the third

body. Using dynamic systems theory, manifolds around each of the Lagrange points of the CR3BP were generated. The stable and unstable manifolds agree with the motion described by the eigenvectors and the motion constraints observed from the zero velocity curves. The manifolds generated have the potential to be used to travel to and from Lagrange points more optimally.

Optimized trajectory solutions for the CR3BP were found using an HJB programming algorithm and the trajectories were compared to the two-dimensional manifold through  $L_1$ . Optimized trajectories were found from an initial position at  $L_1$  to final positions around  $m_2$  for  $\mu = 0.1$ . Cases were run for constrained initial and final positions but with unconstrained final velocities. Additional cases were run with constrained initial and final positions and velocities with the final velocities being forced to circular velocities around  $m_2$ .

The set of cases with the final position of  $(x, y) = (-0.92, 0)$  follow the manifold closer than the other two sets of cases for final positions of  $(x, y) = (-0.91, 0)$  and  $(-0.95, 0)$  throughout the range of times studied. The three data sets studied each have a trajectory with a low time allowance that causes a more direct trajectory to be taken to achieve the final position and does not follow the manifold, resulting in a higher cost. When the time allowance is large, the optimized trajectory leaves the manifold soon after departing  $L_1$  and results in increased distance traveled, which increases the cost required to achieve the final state. The point of increasing cost for a large time allowed was not reached within the time cases run for the final state of  $(x, y) = (-0.92,$

0). This is most likely due to the optimized trajectory following closer to the manifold and the final location being closer to the manifold for the  $(x, y) = (-0.92, 0)$  cases, relative to the cases with final positions of  $(x, y) = (-0.91, 0)$  and  $(-0.95, 0)$ . For the cases with less extreme time allowances, the trajectories stay longer on the manifold and cost less as the endpoint is fixed at a point closer to the manifold and this can be seen in the  $(x, y) = (-0.92, 0)$  cases. The cost in this range of time allowance seems to vary inversely to time allowed when the endpoint is close to the manifold or varies in a more complex manner as the endpoint is fixed farther from the manifold.

These conclusions are the beginning of an area that needs to be further investigated. The influence of manifolds on finding the optimal trajectory for the third body in the CR3BP needs to be studied more to concretely define that manifolds are the optimal pathways to travel in space. This investigation shows that from initial observations, HJB optimized trajectories do use manifolds when finding the optimal pathway to travel in the CR3BP. More research should be done to explore when the optimal trajectory uses a manifold and how to optimize entry onto and exit from the manifold. This paper examined the manifold through  $L_1$ , but other specific manifolds could also be examined to look at a variety of trajectories in the CR3BP. There are many areas of research with manifolds and the CR3BP to explore that would help to better plan for spacecrafts' trajectories in the future. These areas of research have the potential to reduce fuel usage and transit time while exploring new areas of the space frontier.

The better the CR3BP is understood, the more effectively and efficiently humans will be able to explore space.

## 6. Appendices

### *6.1 Appendix A*

```
% Title: rot3body.m
% Description: Program looks at the rotating three-body system
% with 5 libration points.
% This is the circular restricted three-body problem (CR3BP)
% Original Date: 1/9/04
% Author: Raquel Jarabek

clc
clear all
close all
format long

% System has 2 significant massive bodies with a 3rd body moving around
% the other 2 bodies. The 3rd body's mass is assumed zero, because it
% is small enough that its mass does not influence the motion of the 2
% larger bodies. Body 2 is revolving around body 1. Body 3 is free to
% move as instructed. This rotating system creates 5 points where the
% gravitational and centripetal forces are zero.

% This program maps out the force field around the main 2 bodies and
% looks at the movement of the 3rd body.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Choose mu, mass ratio constant

% Definitions: (normalized values)
% m1=1-mu          % largest mass, stationary
% m2=mu           % next largest mass, stationary
% m3=0            % third body, moving
% mu=m2/(m1+m2)   % mass ratio

disp(' ')
disp('The gravitational parameter, mu, is set to:')
%mu= 3.0034e-6    % mass ratio for Sun-Earth system
%mu=0.01213      % mass ratio for the Earth-Moon system; not setup below
%mu=0.5
%mu=0.3
mu=0.1
```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

% Setting up initial equations of motion.

```

```

if mu == 3.0034e-6           % mass ratio for Sun-Earth system
    x=[-1.4:.2:-.2 .2:.2:1.4];
    y=[-1.4:.2:-.2 .2:.3:1.4];
elseif mu == 0.5
    x=[-2:.25:2];
    y=[-2:.25:2];
elseif mu==0.3
    x=[-2.2:.2:2.2];
    y=[-2.2:.2:2.2];
elseif mu==0.1;
%   x=[-2.2:.2:2.2];
%   y=[-2.1:.2:2.1];
    x1=[-2.2:.2:-.2];
    y1=[-2.1:.2:2.1];
    x2=[0:.2:.2];
    y2=[.3:.2:2.1];
    x3=[0:.2:.2];
    y3=[-2.1:.2:-.3];
    x4=[.4:.2:2.2];
    y4=[-2.1:.2:2.1];
else
    x=[-1.5:.25:1.5];
    y=[-1.5:.25:1.5];
    disp('*****ERROR: This mu will not produce the quiver
        plot.*****')
end;

```

```

if mu~=0.1
    [x,y]=meshgrid(x,y);

    % CM12 is the center of mass of bodies 1 and 2 (location of CS origin)
    x1=-mu;           % distance from CM12, to mass 1 (larger mass)
    x2=1-mu;         % distance from CM12, to mass 2 (smaller mass)

    r1=sqrt((x-mu).^2+y.^2);
    r2=sqrt((x+1-mu).^2+y.^2);

    % 3-Body dynamics, rotating system, CR3BP
    % 2 bodies with significant mass (3rd body has no mass)
    x_vel=0;
    y_vel=0;

```



```

% Set velocities to zero to obtain Hill's curves.

om_x=x-((1-mu)*(x-mu))./r1.^3-mu*(x+1-mu)./r2.^3;
om_y=y.*(1-(1-mu))./r1.^3-mu./r2.^3);
x_accel=2*y_vel+om_x;
y_accel=-2*x_vel+om_y;
end;

if mu==0.1
    figure(1)
    hold on
    x_vel=0;
    y_vel=0;

    [x,y]=meshgrid(x1,y1);
    r1=sqrt((x-mu).^2+y.^2);
    r2=sqrt((x+1-mu).^2+y.^2);
    om_x=x-((1-mu)*(x-mu))./r1.^3-mu*(x+1-mu)./r2.^3;
    om_y=y.*(1-(1-mu))./r1.^3-mu./r2.^3);
    x_accel=2*y_vel+om_x;
    y_accel=-2*x_vel+om_y;
    quiver(x,y,x_accel,y_accel,'g')

    [x,y]=meshgrid(x2,y2);
    r1=sqrt((x-mu).^2+y.^2);
    r2=sqrt((x+1-mu).^2+y.^2);
    om_x=x-((1-mu)*(x-mu))./r1.^3-mu*(x+1-mu)./r2.^3;
    om_y=y.*(1-(1-mu))./r1.^3-mu./r2.^3);
    x_accel=2*y_vel+om_x;
    y_accel=-2*x_vel+om_y;
    quiver(x,y,x_accel,y_accel,'g')

    [x,y]=meshgrid(x3,y3);
    r1=sqrt((x-mu).^2+y.^2);
    r2=sqrt((x+1-mu).^2+y.^2);
    om_x=x-((1-mu)*(x-mu))./r1.^3-mu*(x+1-mu)./r2.^3;
    om_y=y.*(1-(1-mu))./r1.^3-mu./r2.^3);
    x_accel=2*y_vel+om_x;
    y_accel=-2*x_vel+om_y;
    quiver(x,y,x_accel,y_accel,'g')

    [x,y]=meshgrid(x4,y4);
    r1=sqrt((x-mu).^2+y.^2);
    r2=sqrt((x+1-mu).^2+y.^2);
    om_x=x-((1-mu)*(x-mu))./r1.^3-mu*(x+1-mu)./r2.^3;
    om_y=y.*(1-(1-mu))./r1.^3-mu./r2.^3);

```

```

x_accel=2*y_vel+om_x;
y_accel=-2*x_vel+om_y;
quiver(x,y,x_accel,y_accel,'g')
end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Quiver plot showing directions of 3rd-body acceleration

if mu~=0.1
    figure(1)
    quiver(x,y,x_accel,y_accel,'g')
    hold on
end;

plot(-1+mu,0,'k',mu,0,'ok') % plotting bodies 1 and 2
title('Quiver plot showing direction of accelerations')
xlabel('x')
ylabel('y')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Libration points locations

% Find all five libration point positions.

% Initialize the x,y,z coordinates of the libration points
L=zeros(5,3); % each row is the location of a libration point
% rows of L = [L1 L2 L3 L4 L5]'
% each column [x y z] (x is 1st column, y 2nd, z
3rd)

% Use pg. 64 of Gomez Vol. 1 to solve for 3 collinear lib. pts. to
% double check the solution found below--done and it checked.

% Find x coordinate of collinear points using Newton's method
% initial values are close to the point you are solving for
clear r1 r2
%x0=[0.0001 1.0001 -1.001];
%x0=[.0001 -1.0001 -.9999];
x0=[mu+.001 mu-1+.001 mu-1-.0001];
for i=1:3
    x=x0(i);
    y=0;

```

```

dx=1;
while abs(dx)>1e-10
    r1=sqrt((x-mu)^2+y^2);
    r2=sqrt((x+1-mu)^2+y^2);

    f=x-(1-mu)*(x-mu)/r1^3-mu*(x+1-mu)/r2^3;

    df_dx=1+2*(1-mu)*invelm(abs(x-mu)^3)+2*mu*invelm(abs(x+1-mu)^3);

    dx=-f/df_dx;
    x=x+dx;
end;
L(i,1)=x;
end;

% Find x and y coordinates of triangular libration points
x_tri=-.5+mu; %-(x1+x2)/2; % from r1=r2=1 & z=0
y_tri=sqrt(3)/2;
L(4,1:2)=[x_tri y_tri];
L(5,1:2)=[x_tri -y_tri];

% Display libration point locations
disp(' ')
disp(' ')
disp('Each row of L is a libration point location = [x y z]')
L

figure(1)
hold on
plot(L(:,1),L(:,2),'r+')

% Plot libration points with labels on separate figure
if mu == 0.1
    figure(2)
    plot(L(:,1),L(:,2),'r+')
    text(L(1,1),L(1,2)-.1,'L_3')
    text(L(2,1),L(2,2)-.1,'L_1')
    text(L(3,1),L(3,2)-.1,'L_2')
    text(L(4,1),L(4,2)-.1,'L_4')
    text(L(5,1),L(5,2)+.08,'L_5')
    hold on
    plot(-1+mu,0,'.b',mu,0,'ob') % plotting bodies 1 and 2
    text(-1+mu,.08,'m_2')
    text(mu,.08,'m_1')
    xlabel('x (LU)')
    ylabel('y (LU)')

```

```

    title('The 5 Libration Points')
end;

% Verify the locations of the libration points found by plugging them back
% into f(x,y) and it should equal zero. f(Lx,Ly)=0 for each point and f is
% the set of EOM below
% x_vel and y_vel are set to zero still

for i=1:5;

    x=L(i,1);
    y=L(i,2);

    fx_accel(i)=2*y_vel+x-((1-mu)*(x-mu))/((x-mu)^2+y^2)^(3/2)-mu*(x+1-
        mu)/((x+1-mu)^2+y^2)^(3/2);
    fy_accel(i)=-2*x_vel+y*(1-(1-mu))/((x-mu)^2+y^2)^(3/2)-mu/((x+1-
        mu)^2+y^2)^(3/2);

end;
fx_accel
fy_accel
disp('If fx_accel and fy_accel = 0, then the libration points are zeros of the equations
    of motion.')
disp(' ')
disp(' ')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Linearize system about each libration point
% Need linearization in order to find eigenvalues and eigenvectors

% del_xdot = A*del_x + B*del_u
% where A=del_f/del_x at libration point
% and B=del_f/del_u at libration point

% f1=x1_dot=x_dot=x3
% f2=x2_dot=y_dot=x4
% f3=x3_dot=x_dot_dot
% f4=x4_dot=y_dot_dot

% Find partial derivatives to complete linearization
clear x1 x2 i
syms de ds x1 x2 x3 x4
% Note: old x1 is now ds and old x2 is now de (ds=old x1=-mu, de=old x2=1-mu)
% Note: old x is now x1 and old y is now x2 (x1=x, x2=y)

```

```

x3_dot=2*x4+x1-
    de*(x1+ds)/((x1+ds)^2+x2^2)^(3/2)+ds*(x1+de)/((x1+de)^2+x2^2)^(3/2);
    % changed signs

A31sym=diff(x3_dot,x1);
%pretty(A31sym);
A32sym=diff(x3_dot,x2);
%pretty(A32sym);

x4_dot=-2*x3+x2*(1-de/((x1+ds)^2+x2^2)^(3/2)+ds/((x1+de)^2+x2^2)^(3/2));    %
    changed signs

A41sym=diff(x4_dot,x1);
%pretty(A41sym);
A42sym=diff(x4_dot,x2);
%pretty(A42sym);

% Substitute real values into A31,A32,A41,A42
A31a=subs(A31sym,de,1-mu);
A31b=subs(A31a,ds,-mu);
A32a=subs(A32sym,de,1-mu);
A32b=subs(A32a,ds,-mu);
A41a=subs(A41sym,de,1-mu);
A41b=subs(A41a,ds,-mu);
A42a=subs(A42sym,de,1-mu);
A42b=subs(A42a,ds,-mu);

for i=1:5;

    A31c(i)=subs(A31b,x1,L(i,1));
    A31(i)=subs(A31c(i),x2,L(i,2));

    A32c(i)=subs(A32b,x1,L(i,1));
    A32(i)=subs(A32c(i),x2,L(i,2));

    A41c(i)=subs(A41b,x1,L(i,1));
    A41(i)=subs(A41c(i),x2,L(i,2));

    A42c(i)=subs(A42b,x1,L(i,1));
    A42(i)=subs(A42c(i),x2,L(i,2));

end;

% Assemble A for the linearization
% del_xdot = A*del_x + B*del_u

```

```

clear i;
i=1;
A1=[0 0 1 0
     0 0 0 1
     A31(i) A32(i) 0 2
     A41(i) A42(i) -2 0];
i=2;
A2=[0 0 1 0
     0 0 0 1
     A31(i) A32(i) 0 2
     A41(i) A42(i) -2 0];
i=3;
A3=[0 0 1 0
     0 0 0 1
     A31(i) A32(i) 0 2
     A41(i) A42(i) -2 0];
i=4;
A4=[0 0 1 0
     0 0 0 1
     A31(i) A32(i) 0 2
     A41(i) A42(i) -2 0];
i=5;
A5=[0 0 1 0
     0 0 0 1
     A31(i) A32(i) 0 2
     A41(i) A42(i) -2 0];

B=[0 0 0 0]';

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Two checks of A matrices (A1,A2,A3,A4,A5) performed to verify that it
% was found correctly.

% A Check #1
% Check A to make sure it was correctly found
%  $f(L+\epsilon V)=f(L)+A*\epsilon V+O(\epsilon^2)$  where  $L=(L_x,L_y)$  and  $f$  is the set of EOM
%  $f(L)=f(L_x,L_y)=0$  was shown in previous section and  $O(\epsilon^2)\sim 0$ 
% so it must be shown that  $f(L+\epsilon V)=A*\epsilon V$ 

clear x1 x2 x3 x4 i

de=1-mu;
ds=-mu;
eps=0.01;

```

```

V=[1 1 1 1]';

for i=1:5;
    Lxy(:,i)=[L(i,1) L(i,2) 0 0]';    % [x y xdot ydot]
    % Lx,Ly=libration point location and there is no initial velocity
    L_eps(:,i)=Lxy(:,i)+eps*V;

    x1(i)=Lxy(1,i);
    x2(i)=Lxy(2,i);
    x3(i)=Lxy(3,i);
    x4(i)=Lxy(4,i);

    f(1,i)=x3(i);
    f(2,i)=x4(i);

    f(3,i)=2*x4(i)+x1(i)-
        de*(x1(i)+ds)/((x1(i)+ds)^2+x2(i)^2)^(3/2)+ds*(x1(i)+de)/((x1(i)+de)^2+x2(i)
        )^2)^(3/2);

    f(4,i)=-2*x3(i)+x2(i)*(1-
        de/((x1(i)+ds)^2+x2(i)^2)^(3/2)+ds/((x1(i)+de)^2+x2(i)^2)^(3/2));

    x1eps(i)=L_eps(1,i);
    x2eps(i)=L_eps(2,i);
    x3eps(i)=L_eps(3,i);
    x4eps(i)=L_eps(4,i);

    f_eps(1,i)=x3eps(i);    % =xdot
    f_eps(2,i)=x4eps(i);    % =ydot

    f_eps(3,i)=2*x4eps(i)+x1eps(i)-
        de*(x1eps(i)+ds)/((x1eps(i)+ds)^2+x2eps(i)^2)^(3/2)+ds*(x1eps(i)+de)/((x1e
        ps(i)+de)^2+x2eps(i)^2)^(3/2);
    f_eps(4,i)=-2*x3eps(i)+x2eps(i)*(1-
        de/((x1eps(i)+ds)^2+x2eps(i)^2)^(3/2)+ds/((x1eps(i)+de)^2+x2eps(i)^2)^(3/2
        ));

end;

f;
f_eps;
AepsV1=A1*eps*V;
AepsV2=A2*eps*V;
AepsV3=A3*eps*V;
AepsV4=A4*eps*V;
AepsV5=A5*eps*V;

```

```

% f_eps=AepsV so A is correct!

% A Check #2
% Check A again using eigenvalues
% A*Vj=lambda*Vj, where lambda=eigenvalue and Vj=eigenvector
% Check for each eigenvector with cooresponding eigenvalue

% Find the eigenvalues and eigenvectors of A
[eigvec1,eigval1]=eig(A1);
[eigvec2,eigval2]=eig(A2);
[eigvec3,eigval3]=eig(A3);
[eigvec4,eigval4]=eig(A4);
[eigvec5,eigval5]=eig(A5);

% for A1
Aeigvec1=A1*eigvec1; % each column is A multiplied by each eigenvector
eigvec_eigval1=eigvec1*eigval1;
% each column is the eigenvalue multiplied by the cooresponding eigenvector

% for A2
Aeigvec2=A2*eigvec2; % each column is A multiplied by each eigenvector
eigvec_eigval2=eigvec2*eigval2;
% each column is the eigenvalue multiplied by the cooresponding eigenvector

% for A3
Aeigvec3=A3*eigvec3; % each column is A multiplied by each eigenvector
eigvec_eigval3=eigvec3*eigval3;
% each column is the eigenvalue multiplied by the cooresponding eigenvector

% for A4
Aeigvec4=A4*eigvec4; % each column is A multiplied by each eigenvector
eigvec_eigval4=eigvec4*eigval4;
% each column is the eigenvalue multiplied by the cooresponding eigenvector

% for A5
Aeigvec5=A5*eigvec5; % each column is A multiplied by each eigenvector
eigvec_eigval5=eigvec5*eigval5;
% each column is the eigenvalue multiplied by the cooresponding eigenvector

% Each Aeigvec=eigvec_eigval so the A's are correct!

%%%%%%%%%%
%%%%%%%%%%

% Plot curves of constant C and zero velocity to get a Hill curve plot

```



```

% works for mu=0.5
[x,y] = meshgrid(-2:.005:2, -2.01:.005:2.01);

% C=2*U, U=potential, V=0; V^2=2*U-C
C=2*(1-mu)./sqrt((x+ds).^2+y.^2) + 2*mu./sqrt((x+de).^2+y.^2) + (1-mu).*((x-
mu).^2+y.^2) + mu.*((x+1-mu).^2+y.^2);

%C = x.^2 + y.^2 + 2*(1-mu)./sqrt((x-ds).^2+y.^2) + 2*mu./sqrt((x-de).^2+y.^2);
% bad C

% draw lines at values, V
L1x=L(1,1);
L1y=L(1,2);
C_L1=2*(1-mu)./sqrt((L1x+ds)^2+L1y^2) + 2*mu./sqrt((L1x+de)^2+L1y^2) + (1-
mu)*((L1x-mu)^2+L1y^2) + mu*((L1x+1-mu)^2+L1y^2);
L2x=L(2,1);
L2y=L(2,2);
C_L2=2*(1-mu)./sqrt((L2x+ds)^2+L2y^2) + 2*mu./sqrt((L2x+de)^2+L2y^2) + (1-
mu)*((L2x-mu)^2+L2y^2) + mu*((L2x+1-mu)^2+L2y^2);
L3x=L(3,1);
L3y=L(3,2);
C_L3=2*(1-mu)./sqrt((L3x+ds)^2+L3y^2) + 2*mu./sqrt((L3x+de)^2+L3y^2) + (1-
mu)*((L3x-mu)^2+L3y^2) + mu*((L3x+1-mu)^2+L3y^2);

if mu == 3.0034e-6 % mass ratio for Sun-Earth system
    V=[1 2 C_L2 4 5];
    %V=[.9 .95 1 1.05 ];
elseif mu == 0.5
    %V=[1 2 3 C_L2 4 C_L1 5];
    V=[1 2 3.1 3.71 4.25 5];
elseif mu==0.3
    %V=[1 1.5 1.917 3 4 5];
    V=[3.01 3.5 3.76 4.13 5];
elseif mu==0.1;
    V=[3.05 C_L1 C_L2 C_L3 5]
% V=[3.05 3.19 3.56 3.69 5];
else
    V=[1 2 3 4 5];
    disp('*****This mu may not produce the best constant velocity
curves.*****')
end;

figure(1)
hold on

```

```

[curve,h]=contour(x,y,C,V,'b');
%h=('C_1')
%clabel(curve,h)
%colorbar

xdata=[0.49999999999450756; 0.44469106895456734; 0.3949525693135281; -
      0.06461803755165016; -0.2999999999766867];
ydata=[0.2999999999689469; 0.09070304003688098; 0.11148357578571826;
      0.12895679351012118; 0.20000000001212503];
%plot(xdata,ydata,'mx-')

title('Curves of zero velocity (C values)')
xlabel('x (LU)')
ylabel('y (LU)')
axis([-2.2 2.2 -2.2 2.2])
axis square

%figure(1)
%surf(x,y,C)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Plot eigenvalues and eigenvectors for each libration point

if mu == 3.0034e-6          % mass ratio for Sun-Earth system
    scale1=1;
    scale2=1;
    scale3=1;
    scale4=1;
elseif mu == 0.5
    scale1=.3;
    scale2=.5;
    scale3=3;
    scale4=.2;
elseif mu==0.3
    scale1=1;
    scale2=1;
    scale3=1;
    scale4=1;
elseif mu==0.1;
    scale1=1;
    scale2=1;
    scale3=1;
    scale4=1;
else

```

```

scale1=1;
scale2=1;
scale3=1;
scale4=1;
disp('*****ERROR: This mu may not produce the best view of the
      eigenvectors.*****')
end;

% eigenvector calculations for L1

x_L1=[L(1,1)+eigvec1(1,1)*scale1      L(1,1)+eigvec1(1,2)*scale3
      (L(1,1)+eigvec1(1,1)+eigvec1(3,1)*scale2)
      (L(1,1)+eigvec1(1,2)+eigvec1(3,2)*scale4)];
y_L1=[L(1,2)+eigvec1(2,1)*scale1      L(1,2)+eigvec1(2,2)*scale3
      (L(1,2)+eigvec1(2,1)+eigvec1(4,1)*scale2)
      (L(1,2)+eigvec1(2,2)+eigvec1(4,2)*scale4)];

im=sqrt(-1);
iter=0;
for theta=0:.1:2*pi;
    iter=iter+1;
    a1(iter)=exp(im*theta);
    a2(iter)=exp(-im*theta);
    xyi_L1(iter,:)=a1(iter)*eigvec1(:,3)+a2(iter)*eigvec1(:,4)';
end;

% eigenvector calculations for L2

x_L2=[L(2,1)+eigvec2(1,1)      L(2,1)+eigvec2(1,2)
      (L(2,1)+eigvec2(1,1)+eigvec2(3,1)) (L(2,1)+eigvec2(1,2)+eigvec2(3,2))];
y_L2=[L(2,2)+eigvec2(2,1)      L(2,2)+eigvec2(2,2)
      (L(2,2)+eigvec2(2,1)+eigvec2(4,1)) (L(2,2)+eigvec2(2,2)+eigvec2(4,2))];

iter=0;
for theta=0:.1:2*pi;
    iter=iter+1;
    a3(iter)=exp(im*theta);
    a4(iter)=exp(-im*theta);
    xyi_L2(iter,:)=a3(iter)*eigvec2(:,3)+a4(iter)*eigvec2(:,4)';
end;

% eigenvector calculations for L3

x_L3=[L(3,1)+eigvec3(1,1)      L(3,1)+eigvec3(1,2)
      (L(3,1)+eigvec3(1,1)+eigvec3(3,1)) (L(3,1)+eigvec3(1,2)+eigvec3(3,2))];
y_L3=[L(3,2)+eigvec3(2,1)      L(3,2)+eigvec3(2,2)

```

```

(L(3,2)+eigvec3(2,1)+eigvec3(4,1)) (L(3,2)+eigvec3(2,2)+eigvec3(4,2));

iter=0;
for theta=0:.1:2*pi;
    iter=iter+1;
    a5(iter)=exp(im*theta);
    a6(iter)=exp(-im*theta);
    xyi_L3(iter,:)=a5(iter)*eigvec3(:,3)+a6(iter)*eigvec3(:,4);
end;

% eigenvector calculations for L4

iter=0;
for theta=0:.1:2*pi;
    iter=iter+1;
    a7(iter)=exp(im*theta);
    a8(iter)=exp(-im*theta);
    xyi_L4a(iter,:)=a7(iter)*eigvec4(:,1)+a8(iter)*eigvec4(:,2);
    xyi_L4b(iter,:)=a7(iter)*eigvec4(:,3)+a8(iter)*eigvec4(:,4);
end;

% eigenvector calculations for L5b

iter=0;
for theta=0:.1:2*pi;
    iter=iter+1;
    a9(iter)=exp(im*theta);
    a10(iter)=exp(-im*theta);
    xyi_L5a(iter,:)=a9(iter)*eigvec5(:,1)+a10(iter)*eigvec5(:,2);
    xyi_L5b(iter,:)=a9(iter)*eigvec5(:,3)+a10(iter)*eigvec5(:,4);
end;

% Plots for all eigenvectors of all 5 libration points

% plot(x_L1(:,1),y_L1(:,1),'m-',x_L1(2,1),y_L1(2,1),'m.')
% plot(-x_L1(:,1)+2*L(1,1),-y_L1(:,1),'m-',-x_L1(2,1)+2*L(1,1),-y_L1(2,1),'m.')
% plot(x_L1(:,2),y_L1(:,2),'m-',x_L1(2,2),y_L1(2,2),'m.')
% plot(-x_L1(:,2)+2*L(1,1),-y_L1(:,2),'m-',-x_L1(2,2)+2*L(1,1),-y_L1(2,2),'m.')

%plot(xyi_L1(:,1)+L(1,1),xyi_L1(:,2),'y')

% plot(x_L2(:,1),y_L2(:,1),'m-',x_L2(2,1),y_L2(2,1),'m.')
% plot(-x_L2(:,1)+2*L(2,1),-y_L2(:,1),'m-',-x_L2(2,1)+2*L(2,1),-y_L2(2,1),'m.')
% plot(x_L2(:,2),y_L2(:,2),'m-',x_L2(2,2),y_L2(2,2),'m.')
% plot(-x_L2(:,2)+2*L(2,1),-y_L2(:,2),'m-',-x_L2(2,2)+2*L(2,1),-y_L2(2,2),'m.')
% %plot(xyi_L2(:,1)+L(2,1),xyi_L2(:,2),'y')

```

```

%
% plot(x_L3(:,1),y_L3(:,1),'m-',x_L3(2,1),y_L3(2,1),'m.')
% plot(-x_L3(:,1)+2*L(3,1),-y_L3(:,1),'m-',-x_L3(2,1)+2*L(3,1),-y_L3(2,1),'m.')
% plot(x_L3(:,2),y_L3(:,2),'m-',x_L3(2,2),y_L3(2,2),'m.')
% plot(-x_L3(:,2)+2*L(3,1),-y_L3(:,2),'m-',-x_L3(2,2)+2*L(3,1),-y_L3(2,2),'m.')
% %plot(xyi_L3(:,1)+L(3,1),xyi_L3(:,2),'y')

% plot(xyi_L4a(:,1)+L(4,1),xyi_L4a(:,2)+L(4,2),'c')
% plot(xyi_L4b(:,1)+L(4,1),xyi_L4b(:,2)+L(4,2),'c')
%
% plot(xyi_L5a(:,1)+L(5,1),xyi_L5a(:,2)+L(5,2),'c')
% plot(xyi_L5b(:,1)+L(5,1),xyi_L5b(:,2)+L(5,2),'c')

%%%%%%%%%%
%%%%%%%%%%

% Equations of motion used to propagate forwards in time

% These values are used to create the figure-8 through L1 between m1 and m2.
if mu == 0.1
    % for L1--row 2
    tp1 = 2.3826; % time to propagate forwards and backwards integration of EOM
    near m1
    tp2 = 2.853; % time to propagate forwards and backwards integration of EOM
    near m2
%   tp1 = 5; % time to propagate forwards and backwards integration of EOM
%   near m1
%   tp2 = 8; % time to propagate forwards and backwards integration of EOM
%   near m2
    % for L3--row 1
%   tp1 = 16.92; % time to propagate forwards and backwards integration of EOM
%   near m1
%   tp2 = 19.8; % time to propagate forwards and backwards integration of EOM
%   near m2
    % for L2--row 3
%   tp1 = 10; % time to propagate forwards and backwards integration of EOM
%   near m1
%   tp2 = 13; % time to propagate forwards and backwards integration of EOM
%   near m2
    % for L4--row 4
%   tp1 = 40; % time to propagate forwards and backwards integration of EOM
%   near m1
%   tp2 = 40; % time to propagate forwards and backwards integration of EOM
%   near m2
elseif mu == 0.5

```

```

    tp1 = 2.331;    % time to propagate forwards and backwards integration of EOM
                    near m1
    tp2 = 2.331;    % time to propagate forwards and backwards integration of EOM
                    near m2
else
    disp('*****ERROR: This mu is not setup to produce the correct
        manifold.*****')
end;

% rv0for=[x y xdot ydot] initial location
%rv0for=[eigvec1(1,1)*.0001 eigvec1(2,1)*.0001 eigvec1(3,1)*0 eigvec1(4,1)*0];

rv0for1=[L(2,1)-.0001 L(2,2)+.0001 0 0];
%rv0for1=[L(1,1)-.0001 L(1,2)+.0001 0 0];
%rv0for1=[L(3,1)-.0001 L(3,2)+.0001 0 0];
%rv0for1=[L(4,1)-.0001 L(4,2)+.0001 0 0];

x=rv0for1(1);
y=rv0for1(2);
xdot=rv0for1(3);
ydot=rv0for1(4);
C1a=-xdot^2-ydot^2+2*(1-mu)./sqrt((x+ds).^2+y.^2) + 2*mu./sqrt((x+de).^2+y.^2)
    + (1-mu).*((x-mu).^2+y.^2) + mu.*((x+1-mu).^2+y.^2);

[t_for1,rv1for1]=ode45('forward_rot_eom',[0 tp1],rv0for1,odeset('RelTol',1e-
    12,'AbsTol',1e-12),mu);
[m,n]=size(rv1for1);
x=rv1for1(m,1);
y=rv1for1(m,2);
xdot=rv1for1(m,3);
ydot=rv1for1(m,4);
C1b=-xdot^2-ydot^2+2*(1-mu)./sqrt((x+ds).^2+y.^2) + 2*mu./sqrt((x+de).^2+y.^2)
    + (1-mu).*((x-mu).^2+y.^2) + mu.*((x+1-mu).^2+y.^2);

% p=3000;
% x=rv1for1(p,1);
% y=rv1for1(p,2);
% xdot=rv1for1(p,3);
% ydot=rv1for1(p,4);
% C1c=-xdot^2-ydot^2+2*(1-mu)./sqrt((x+ds).^2+y.^2) +
    2*mu./sqrt((x+de).^2+y.^2) + (1-mu).*((x-mu).^2+y.^2) + mu.*((x+1-
    mu).^2+y.^2);

rv0for2=[L(2,1)+.0001 L(2,2)-.0001 0 0];
%rv0for2=[L(1,1)+.0001 L(1,2)-.0001 0 0];
%rv0for2=[L(3,1)+.0001 L(3,2)-.0001 0 0];

```

```

%rv0for2=[L(4,1)+.0001 L(4,2)-.0001 0 0];

x=rv0for2(1);
y=rv0for2(2);
xdot=rv0for2(3);
ydot=rv0for2(4);
C2a=-xdot^2-ydot^2+2*(1-mu)./sqrt((x+ds).^2+y.^2) + 2*mu./sqrt((x+de).^2+y.^2)
      + (1-mu).*((x-mu).^2+y.^2) + mu.*((x+1-mu).^2+y.^2);

[t_for2,rv1for2]=ode45('forward_rot_eom',[0 tp2],rv0for2,odeset('RelTol',1e-
      12,'AbsTol',1e-12),mu);
[m,n]=size(rv1for2);
x=rv1for2(m,1);
y=rv1for2(m,2);
xdot=rv1for2(m,3);
ydot=rv1for2(m,4);
C2b=-xdot^2-ydot^2+2*(1-mu)./sqrt((x+ds).^2+y.^2) + 2*mu./sqrt((x+de).^2+y.^2)
      + (1-mu).*((x-mu).^2+y.^2) + mu.*((x+1-mu).^2+y.^2);

plot(rv1for1(:,1),rv1for1(:,2),'k')%,rv0for1(1),rv0for1(2),'*')
plot(rv1for2(:,1),rv1for2(:,2),'k')%,rv0for2(1),rv0for2(2),'*')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Equations of motion used to propagate backwards in time

% rv0bac=[x y xdot ydot] initial location
rv0bac1=[L(2,1)+.0001 L(2,2)+.0001 0 0];
%rv0bac1=[L(1,1)+.0001 L(1,2)+.0001 0 0];
%rv0bac1=[L(3,1)+.0001 L(3,2)+.0001 0 0];
%rv0bac1=[L(4,1)+.0001 L(4,2)+.0001 0 0];

[t_bac1,rv1bac1]=ode45('backward_rot_eom',[0 tp2],rv0bac1,odeset('RelTol',1e-
      12,'AbsTol',1e-12),mu);

rv0bac2=[L(2,1)-.0001 L(2,2)-.0001 0 0];
%rv0bac2=[L(1,1)-.0001 L(1,2)-.0001 0 0];
%rv0bac2=[L(3,1)-.0001 L(3,2)-.0001 0 0];
%rv0bac2=[L(4,1)-.0001 L(4,2)-.0001 0 0];

[t_bac2,rv1bac2]=ode45('backward_rot_eom',[0 tp1],rv0bac2,odeset('RelTol',1e-
      12,'AbsTol',1e-12),mu);

plot(rv1bac1(:,1),rv1bac1(:,2),'r')%,rv0bac1(1,1),rv0bac1(1,2),'*')
plot(rv1bac2(:,1),rv1bac2(:,2),'r')%,rv0bac2(1,1),rv0bac2(1,2),'*')

```

```

figure(1)
if mu == 0.1
    title('Circular Restricted 3-Body Problem with Hill Curves of Zero Velocity,
          mu=0.1')
elseif mu == 0.5
    title('Circular Restricted 3-Body Problem with Hill Curves of Zero Velocity,
          mu=0.5')
else
    title('Circular Restricted 3-Body Problem with Hill Curves of Zero Velocity')
end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% for mu=0.1; this can be run to compare to results.txt and results2.txt
% rv0new=[-0.91 0 1 1];
% [tnew,rvnew]=ode45('forward_rot_eom',[0 .5],rv0new,odeset('RelTol',1e-
    12,'AbsTol',1e-12),mu);
% plot(rvnew(:,1),rvnew(:,2),'m')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% This case is to verify EOM and the manifold curve through L1 for mu=0.5

if mu == 0.5
    % Load comparison data from Michel Santos
    load time.dat
    load positionx.dat
    load positiony.dat
    load speedx.dat
    load speedy.dat

    p=1;
    x=rv1for1(p,1);
    y=rv1for1(p,2);
    xdot=rv1for1(p,3);
    ydot=rv1for1(p,4);
    C_check=-xdot^2-ydot^2+2*(1-mu)/sqrt((x+ds).^2+y.^2) +
        2*mu./sqrt((x+de).^2+y.^2) + (1-mu).*((x-mu).^2+y.^2) + mu.*((x+1-
        mu).^2+y.^2);

    % Plot data
    figure(3)
    plot(positionx,positiony,'b')

```



```

end;

% Plotting data from results.txt
if mu == 0.1
    load xydata1.dat
    load xydata2.dat
    load xydata3.dat
    load xydata5.dat
    load xydata6.dat
    load xydata7.dat
    load xydata8.dat
    load xydata9.dat
    load xydata10.dat
    load xydata11.dat

    x1=xydata1(1,:);
    y1=xydata1(2,:);
    x2=xydata2(1,:);
    y2=xydata2(2,:);
    x3=xydata3(1,:);
    y3=xydata3(2,:);
    x5=xydata5(1,:);
    y5=xydata5(2,:);
    x6=xydata6(1,:);
    y6=xydata6(2,:);
    x7=xydata7(1,:);
    y7=xydata7(2,:);
    x8=xydata8(1,:);
    y8=xydata8(2,:);
    x9=xydata9(1,:);
    y9=xydata9(2,:);
    x10=xydata10(1,:);
    y10=xydata10(2,:);
    x11=xydata11(1,:);
    y11=xydata11(2,:);

    L1x=-0.60903511002320;    % x location of L1
    L1y=0;                    % y location of L1
    m1x=.1;                   % x location of mass 1
    m1y=0;                    % y location of mass 1
    m2x=-.9;                  % x location of mass 2
    m2y=0;                    % y location of mass 2

    figure(3)
    hold on
    plot(x1,y1,'r',x2,y2,'g',x3,y3,'c',x5,y5,'b',x7,y7,'--r',...

```

```

x8,y8,'--g',x9,y9,'--c',x10,y10,'--b',x11,y11,'--m',...
rv1for1(:,1),rv1for1(:,2),'k',rv1bac2(:,1),rv1bac2(:,2),'k')

    legend('.25TU','.5TU','.75TU','1.0TU','1.25TU','1.5TU','1.75TU','2.0TU','2.6T
    U','manifold',4)
plot(L1x,L1y,'*b',m2x,m2y,'ob')
title('Manifold with Circular End Velocity HJB Optimization Results for:')
text(-.87,.14,'(x_o,y_o)=(L_1); (x_f,y_f)=(-0.91,0); mu=0.1')
xlabel('x (LU)')
ylabel('y (LU)')
axis([-0.97 -0.58 -0.2 0.15])
text(-.89,0,'m_2')
text(-.6,0,'L_1')

figure(4)
hold on
plot(x6,y6,'r--',...
    rv1for1(:,1),rv1for1(:,2),'k',rv1bac2(:,1),rv1bac2(:,2),'k')
legend('1.0TU unconstrained','manifold',4)
plot(L1x,L1y,'*b',m2x,m2y,'ob')
title('Manifold with Unconstrained End Velocity HJB Optimization Result for:')
text(-.87,.14,'(x_o,y_o)=(L_1); (x_f,y_f)=(-0.91,0); mu=0.1')
xlabel('x (LU)')
ylabel('y (LU)')
axis([-0.97 -0.58 -0.2 0.15])
text(-.89,0,'m_2')
text(-.6,0,'L_1')

% Plotting data from results4.txt

load xy4data1.dat
load xy4data2.dat
load xy4data3.dat
load xy4data4.dat
load xy4data5.dat
load xy4data6.dat
load xy4data7.dat
load xy4data8.dat

x4_1=xy4data1(1,:);
y4_1=xy4data1(2,:);
x4_2=xy4data2(1,:);
y4_2=xy4data2(2,:);
x4_3=xy4data3(1,:);
y4_3=xy4data3(2,:);
x4_4=xy4data4(1,:);

```

```

y4_4=xy4data4(2,:);
x4_5=xy4data5(1,:);
y4_5=xy4data5(2,:);
x4_6=xy4data6(1,:);
y4_6=xy4data6(2,:);
x4_7=xy4data7(1,:);
y4_7=xy4data7(2,:);
x4_8=xy4data8(1,:);
y4_8=xy4data8(2,:);

figure(5)
hold on
plot(x4_1,y4_1,'r',x4_2,y4_2,'g',x4_3,y4_3,'c',x4_4,y4_4,'b',x4_5,y4_5,'m',...
     x4_6,y4_6,'r--',x4_7,y4_7,'g--',x4_8,y4_8,'c--',...
     rv1for1(:,1),rv1for1(:,2),'k',rv1bac2(:,1),rv1bac2(:,2),'k')
legend('.5TU','1.0TU','1.5TU','2.0TU','2.5TU','3.0TU','3.5TU','4.0TU','manifold',4)
plot(L1x,L1y,'*b',m2x,m2y,'ob')
title('Manifold with Unconstrained End Velocity HJB Optimization Results for:')
text(-.87,.14,'(x_o,y_o)=(L_1); (x_f,y_f)=(-0.92,0); mu=0.1')
xlabel('x (LU)')
ylabel('y (LU)')
axis([- .97 -.58 -.2 .15])
text(-.89,0,'m_2')
text(-.6,0,'L_1')

% Plotting data from results3.txt

load xy3data1.dat
load xy3data2.dat
load xy3data3.dat
load xy3data4.dat
load xy3data5.dat
load xy3data6.dat

x3_1=xy3data1(1,:);
y3_1=xy3data1(2,:);
x3_2=xy3data2(1,:);
y3_2=xy3data2(2,:);
x3_3=xy3data3(1,:);
y3_3=xy3data3(2,:);
x3_4=xy3data4(1,:);
y3_4=xy3data4(2,:);
x3_5=xy3data5(1,:);
y3_5=xy3data5(2,:);
x3_6=xy3data6(1,:);
y3_6=xy3data6(2,:);

```

```

figure(6)
hold on
plot(x3_1,y3_1,'r',x3_2,y3_2,'g',x3_3,y3_3,'c',x3_4,y3_4,'b',x3_5,y3_5,'m',...
     rv1for1(:,1),rv1for1(:,2),'k',rv1bac2(:,1),rv1bac2(:,2),'k')
legend('.5TU','1.0TU','1.5TU','2.0TU','2.5TU','manifold',4)
plot(L1x,L1y,'*b',m2x,m2y,'ob')
title('Manifold with Unconstrained End Velocity HJB Optimization Results for:')
text(-.87,.14,'(x_o,y_o)=(L_1); (x_f,y_f)=(-0.95,0); mu=0.1')
xlabel('x (LU)')
ylabel('y (LU)')
axis([- .97 - .58 - .2 .15])
text(-.89,0,'m_2')
text(-.6,0,'L_1')

figure(7)
hold on
plot(x3_5,y3_5,'b',x3_6,y3_6,'m--',...
     rv1for1(:,1),rv1for1(:,2),'k',rv1bac2(:,1),rv1bac2(:,2),'k')
legend('2.5TU normal guess','2.5TU looped guess','manifold',4)
plot(L1x,L1y,'*b',m2x,m2y,'ob')
title('Manifold with Unconstrained End Velocity HJB Optimization Results for:')
text(-.87,.14,'(x_o,y_o)=(L_1); (x_f,y_f)=(-0.95,0); mu=0.1')
xlabel('x (LU)')
ylabel('y (LU)')
axis([- .97 - .58 - .2 .15])
text(-.89,0,'m_2')
text(-.6,0,'L_1')

end;

```

## 7.2 Appendix B

```

function Xdot=forward_rot_eom(t,x,flag,mu)

% function Xdot=forward_rot_eom(t,x,flag,input)
%
% 2-Dimensional Nonlinear State Equations for the X-Y Plane
% Simple 3-body rotating system
% EOM with 5 libration points
% This integration goes forward in time.
%
% t=integration initial and final times
% x=[x y xdot ydot]=[x1 x2 x3 x4] IC's
% mu=mass ratio

```

```

%
% written by Raquel Jarabek
% date: 7/22/03

de=1-mu;    % distance from CM (origin) to smaller body (Earth)
ds=-mu;    % distance from CM (origin) to larger body (Sun)

Xdot=[x(3)
      x(4)
      2*x(4)+x(1)-
        de*(x(1)+ds)/((x(1)+ds)^2+x(2)^2)^(3/2)+ds*(x(1)+de)/((x(1)+de)^2+x(2)^2)
        ^3/2)
      -2*x(3)+x(2)*(1-de/((x(1)+ds)^2+x(2)^2)^(3/2)+ds/((x(1)+de)^2+x(2)^2)^(3/2))];

```

### 6.3 Appendix C

```

function Xdot=backward_rot_eom(t,x,flag,mu)

% function Xdot=backward_rot_eom(t,x,flag,input)
%
% 2-Dimensional Nonlinear State Equations for the X-Y Plane
% Simple 3-body rotating system
% EOM with 5 libration points
% This integrations goes backwards in time and is done
% by substituting -x2 for x2.
%
% t=integration initial and final times
% x=[x y xdot ydot]=[x1 x2 x3 x4] IC's
% mu=mass ratio
%
% written by Raquel Jarabek
% date: 1/29/04

de=1-mu;    % distance from CM (origin) to Earth
ds=-mu;    % distance from CM (origin) to Sun

Xdot=[x(3)
      x(4)
      -2*x(4)+x(1)-
        de*(x(1)+ds)/((x(1)+ds)^2+x(2)^2)^(3/2)+ds*(x(1)+de)/((x(1)+de)^2+x(2)^2)
        ^3/2)
      2*x(3)+x(2)*(1-de/((x(1)+ds)^2+x(2)^2)^(3/2)+ds/((x(1)+de)^2+x(2)^2)^(3/2))];

```

#### 6.4 Appendix D

```
function xm=invelm(x)
% function xm=invelm(x)
%
% Calculates the element by element inverse of a matrix.
%
% B. Keller, 1/8/95

if length(x)==1, xm=1/x;
else, [i,j]=size(x); xm=ones(i,j)./x; end
```

## 7. References

- Bellman, R. (1957). *Dynamic Programming*, Princeton University Press, Princeton, NJ.
- Bhatti, M. A. (1998). *Practical Optimization Methods*, Springer, New York, NY.
- Dorato, P., Abdallah, C., and Cerone, V. (1995). *Linear-Quadratic Control: An Introduction*, Prentice Hall, Upper Saddle River, NJ.
- Epperson, J. T. (2002). *An Introduction to Numerical Methods and Analysis*, John Wiley & Sons, Inc., New York, NY.
- Gomez, G., Jorba, A., Masdemont, J., and Simo, C. (1993). "Study Of The Transfer From The Earth To A Halo Orbit Around The Equilibrium Point L1." *Celestial Mechanics and Dynamical Astronomy*, 56, 541-562.
- Gomez, G., and Masdemont, J. (2000). "Some Zero Cost Transfers Between Libration Point Orbits." *Advances in the Astronautical Sciences*, 105, 1199-1216.
- Howell, K. C., Barden, B. T., and Lo, M. W. (1997a). "Application of dynamical systems theory to trajectory design for a libration point mission." *Journal of the Astronautical Sciences*, 45(2), 161-178.
- Howell, K. C., Barden, B. T., Wilson, R. S., and Lo, M. W. (1997b). "Trajectory Design Using a Dynamical Systems Approach With Application to Genesis." *Advances in the Astronautical Sciences*, 97(2), 1665-1684.
- Howell, K. C., Mains, D. L., and Barden, B. T. (1994). "Transfer Trajectories From Earth Parking Orbits To Sun-Earth Halo Orbits." *Advances in the Astronautical Sciences*, 87(1), 399-422.
- Lo, M. W. (1997). "Halo Orbit Generation Using The Center Manifold." *Advances in the Astronautical Sciences*, 95(1), 109-116.
- Lo, M. W., and Ross, S. D. (1998). "Low Energy Interplanetary Transfers Using The Invariant Manifolds Of L1, L2, And Halo Orbits." *Advances in the Astronautical Sciences*, 99(1), 551-561.
- Roy, A. E. (1965). *The Foundations Of Astrodynamics*, The Macmillan Company, New York, NY.
- Serban, R., Koon, W. S., Lo, M. W., Marsden, J. E., Petzold, L. R., Ross, S. D., and Wilson, R. S. (2002). "Halo orbit mission correction maneuvers using optimal control." *Automatica*, 38(4), 571-583.
- Starchville, T. F., Jr., and Melton, R. G. (1998). "Optimal Low-Thrust Transfers To Halo Orbits About The L2 Libration Point In The Earth-Moon System (Elliptical Problem)." *Advances in the Astronautical Sciences*, 99(2), 1489-1505.
- Stoer, J., and Bulirsch, R. (1993). *Introduction to Numerical Analysis*, Springer, New York, NY.
- Szebehely, V. (1967). *Theory of Orbits: The Restricted Problem of Three Bodies*, Academic Press, New York, NY.
- von Stryk, O., and Schlemmer, M. (1994). "Optimal Control of the Industrial Robot Manutec r3." *International Series of Numerical Mathematics*, 115, 367-382.
- Wiesel, W. E. (1989). *Spaceflight Dynamics*, McGraw-Hill Publishing Company, New York, NY.