ASEN 6060 - HW 4

Spring 2025

Jash Bhalavat

Problem 1 - Part a

i) Firstly, let's reference the position of the L2 equilibrium point from HW 2:

Equilibrium Point	x	y	z
L2	**1.155682160772215**	**0**	**0**

Then, the in-plane mode can be calculated using the following equation:

$$\lambda = \pm i\sqrt{U^*_{zz}\big|_{x_{eq}}} \qquad \Lambda = \lambda^2 \qquad \Lambda = \frac{-4 + U^*_{xx} + U^*_{yy}}{2} \pm \frac{\sqrt{(4 - U^*_{xx} - U^*_{yy})^2 - 4(U^*_{xx}U^*_{yy} - (U^*_{xy})^2)}}{2}$$

Each equilibrium point will have 6 modes (4x in-plane and 2x out-of-plane). Only the in-plane modes are necessary for this problem to get initial conditions for the L2 Lyapunov orbit family. The in-plane modes are as follows:

In Plane Modes	
Equilibrium Point	**Eigenvalues**
L2	• λ_1 = **2.158674331407204 + 0i** • λ_2 = **-2.158674331407204 + 0i** • λ_3 = **0.0 + 1.862645868650095i** • λ_4 = **0.0 - 1.862645868650095i**

For a periodic orbit, the eigenvectors of eigenvalues of form +/-bi are used i.e. λ_3, λ_4. Eigenvectors for λ_3, λ_4 can be calculated by generating the A2D matrix and using Matlab's eig() function. The A2D matrix for L2 can be generated using the following form:

$$\begin{bmatrix} \dot{\xi} \\ \dot{\eta} \\ \ddot{\xi} \\ \ddot{\eta} \end{bmatrix} = \begin{bmatrix} 0_{2\text{x}2} & I_{2\times 2} \\ U^*_{XX} & \Omega \end{bmatrix} \begin{bmatrix} \xi \\ \eta \\ \dot{\xi} \\ \dot{\eta} \end{bmatrix} \longrightarrow \delta\dot{\bar{x}}_{2D} = A_{2D}\delta\bar{x}_{2D}$$

Using Matlab to find the eigenvectors:

- Eigenvector for λ_3:
  - [0.153601013997998 + 0i, 0.0 + 0.447378948016456i, 0.0 + 0.286104294143837i, -0.833308549243876 + 0i]
  - Real components
    - [0.153601013997998, 0.0, 0.0, -0.833308549243876]
  - Imaginary components (i is assumed)
    - [0, 0.447378948016456, 0.286104294143837, 0]
- Eigenvector for λ_4:
  - [0.153601013997998 + 0i, 0.0 - 0.447378948016456i, 0.0 - 0.286104294143837i, -0.833308549243876 + 0i]
  - Real components
    - [0.153601013997998, 0.0, 0.0, -0.833308549243876]
  - Imaginary components (i is assumed)
    - [0, -0.447378948016456, -0.286104294143837, 0]

The real and imaginary components of the complex conjugate pair:

- Real components
  - u = [0.153601013997998, 0.0, 0.0, -0.833308549243876]
- Imaginary components
  - v = [0, 0.447378948016456, 0.286104294143837, 0]

u and v form a set of real basis vectors to find initial guesses for the L2 Lyapunov orbit. A linear combination of u, v can be used to find initial guesses for a periodic path around L2 (let's call this dx0):

dx0 = n*u + m*v (where m and n are real numbers)

For this problem let m = 0, and n = 6.510373688112712e-04. So:
dx0 = [0.001, 0, 0, -0.542515005307671]

Hence, the vector formed by linearly combining u and v can serve as an initial guess for a periodic path around L2 in this linear system. The initial condition can be found by adding this initial perturbation to L2's position.

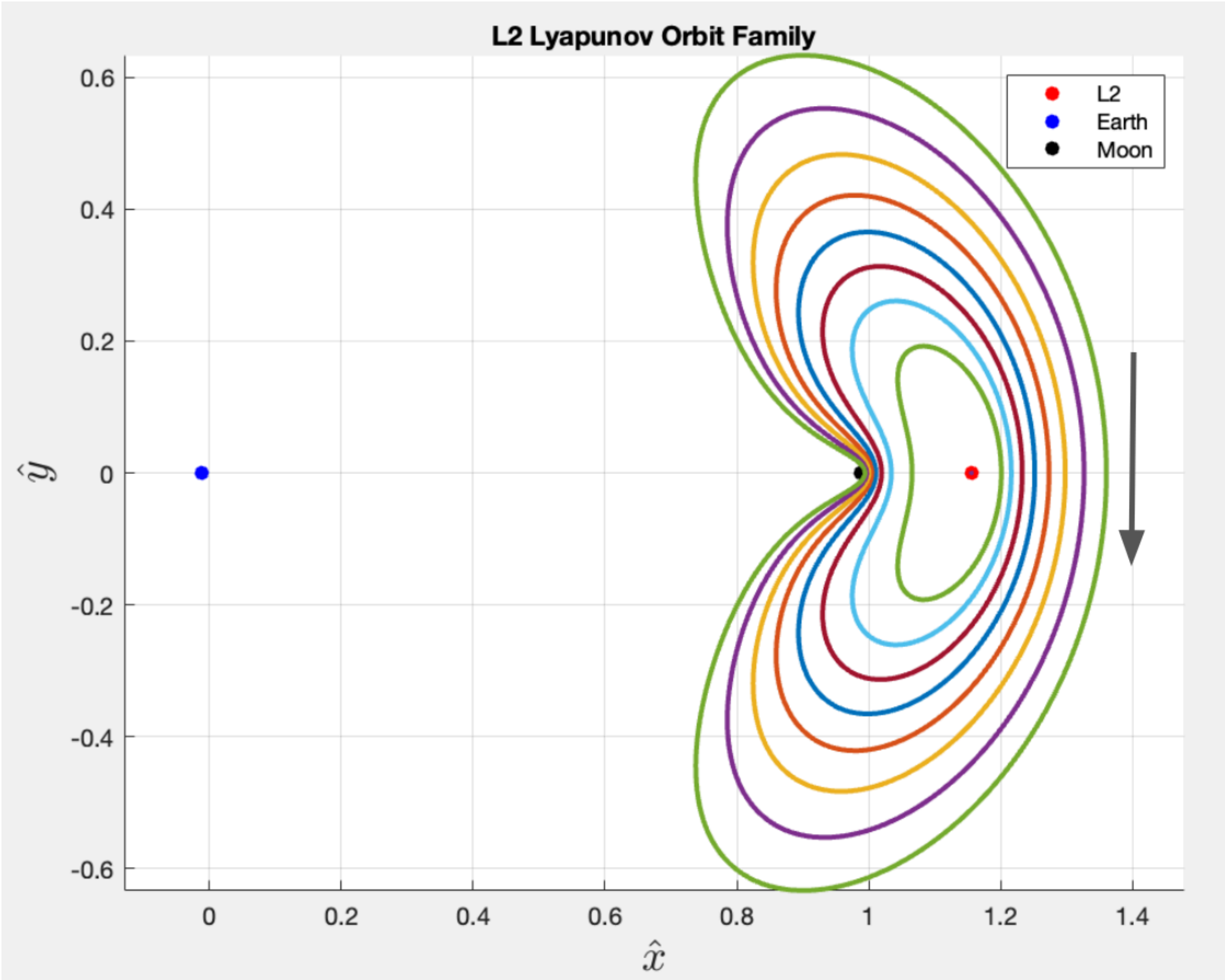
- $\bar{x}_0 = [1.155782160772215, 0,\ 0,\ 0,\ \ -0.000542515005308,\ 0]$ **(unitless)**
- **Period (found by dividing the imaginary part of the eigenvalue by 2*π) - 3.373258123259450 unitless**

Then, using the corrections and continuation scheme from HW 3, the L2 Lyapunov Orbit family is computed. The initial step size used for this orbit is -0.01 [-]. From HW3, the majority of the corrections and continuation scheme remains the same. The biggest changes are listed below:

- An adaptive step size is implemented with the following logic:
    - For every new family member, an initial step size is first used (differs for families, but for L2 Lyapunov orbit it is -0.01 [-]).
    - This step size is used to calculate all the necessary matrices. If there is an error during any of these calculations (I was mostly seeing singular matrix warnings), those errors are then caught using Matlab's try and catch logic and the step size is halved.
    - This is repeated until there are no errors/warnings from the computations or if this loop is repeated 10 times.
- For HW3, only the initial and final points (which are the same) were used to check if they hit the earth or moon's surface. But, now, every new family member is propagated and each point is checked to see if it hits the earth or moon's surface. If it does, the orbit family continuation is terminated.

Apart from these, tolerances, ODE schemes and matrix/vector computations remain the same as HW3.

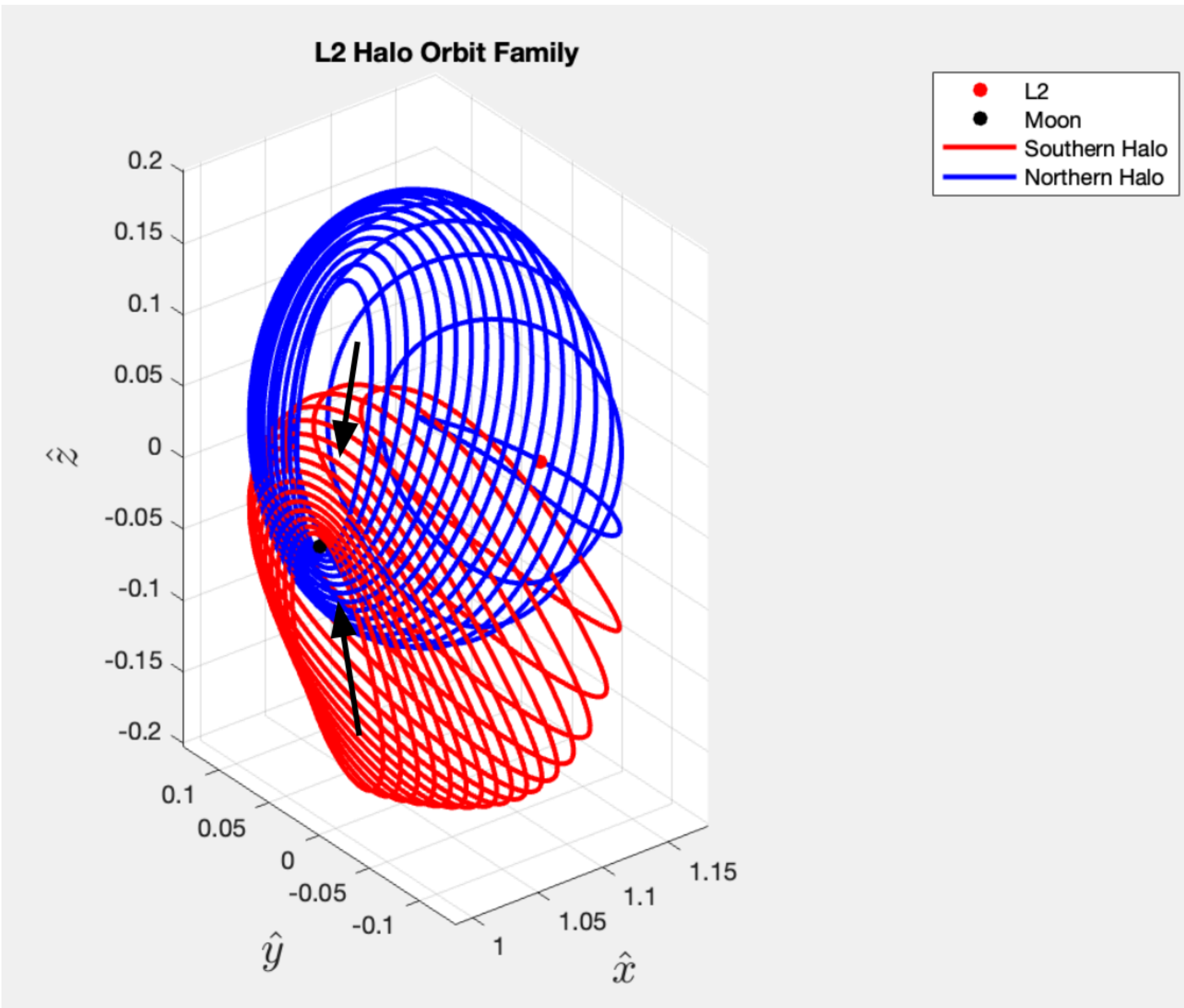
Using this updated continuation scheme, the L2 Lyapunov orbit family is calculated and plotted below:



L2 Lyapunov Orbit Family
L2
Earth
Moon
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
0
0.2
0.4
0.6
0.8
1
1.2
1.4
$\hat{x}$
$\hat{y}$

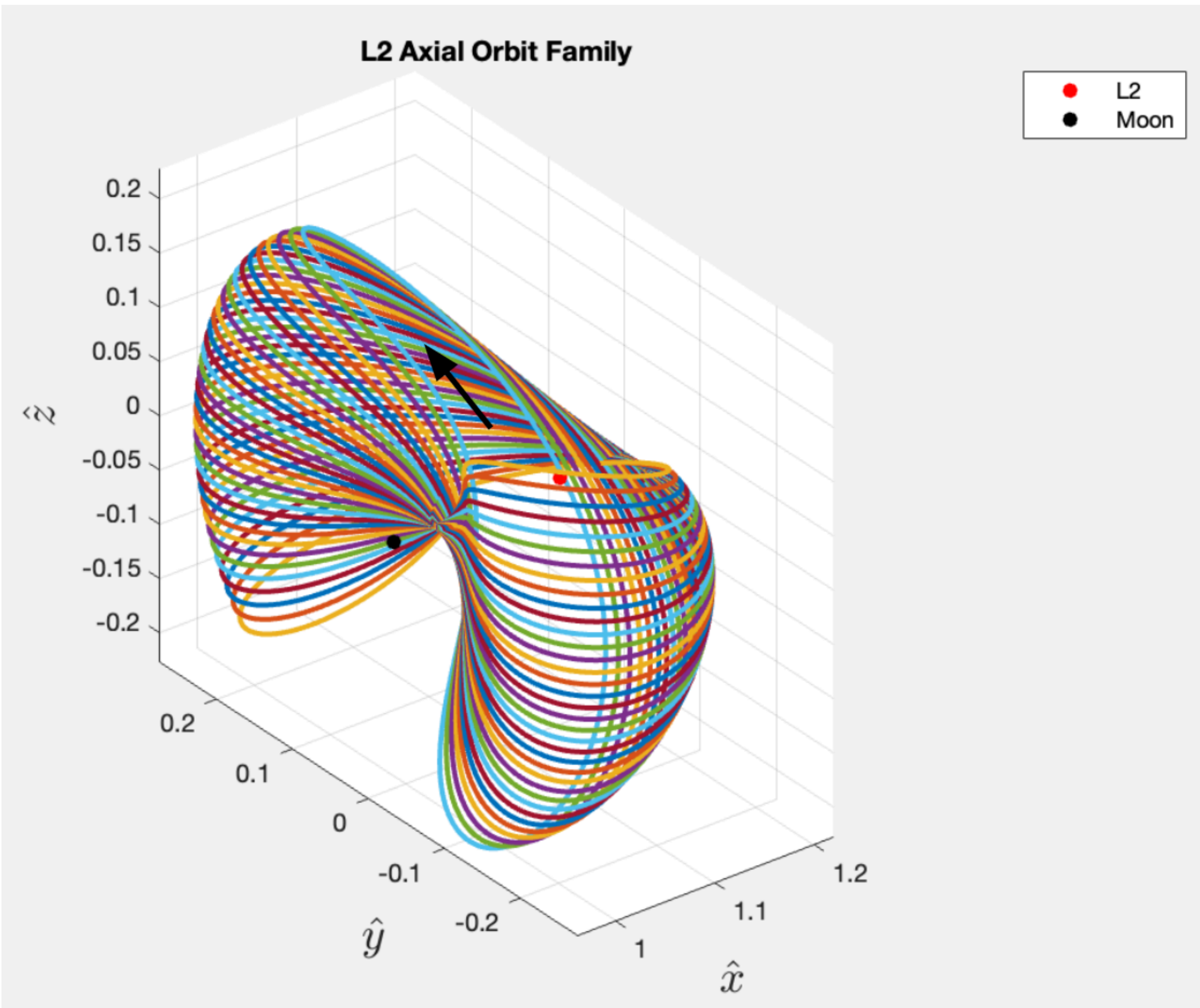
ii) Using the initial condition provided, the L2 halo orbit family is calculated and plotted below. The southern halo orbit family is computed using an initial step size of -0.0005 [-]. Then, the step size is changed to 0.0005 [-] to get the northern halo orbit family:

(The moon is right at the intersection of of the northern and southern orbits and is a small black dot that hides behind the intersecting orbits)



iii) Using the initial condition provided, the L2 axial orbit family is calculated and plotted below:

(This orbit family is discontinuous for me. I have tried positive and negative step sizes and the vertical orbits seem to always cause matrix singularity in my implementation. Hence, there is a discontinuity in the orbit family. I have tried various step sizes and)



The step size used for the computation of this family is 0.001 and -0.001[-].

Problem 1 - Part b

L2 Halo Orbit (first member)

- Monodromy matrix
    - The monodromy matrix is found by simply numerically integrating the initial conditions using the 42x1 differential equation and converting the last row to the state transition matrix. For the L2 Halo orbit, the monodromy matrix is as follows:

```
1.0e+03 *

 0.6791   -0.2558   -0.0503    0.2628    0.0925   -0.0108
-0.2396    0.0910    0.0179   -0.0925   -0.0328    0.0038
-0.0279    0.0105    0.0031   -0.0108   -0.0038    0.0003
 1.2767   -0.4801   -0.0946    0.4941    0.1740   -0.0203
-0.6960    0.2626    0.0516   -0.2697   -0.0939    0.0112
-0.1304    0.0490    0.0097   -0.0503   -0.0179    0.0031
```

(halo monodromy matrix with higher decimal points can be found at the end of this problem)

- Eigenvalues and Eigenvectors

Name (eval)	Eigenvalue	Name (evec)	Eigenvector	Stability
λ_1	**1.172424991777127e+03**	v_1	**[-0.417029338634838 0.147020584656656 0.017156612314002 -0.783934450142028 0.428050146573006 0.079981116260377]**	**Unstable Mode**
λ_2	**0.000000852933029e+03**	v_2	**[-0.417029338635101 -0.147020584656995 0.017156612313994 0.783934450141285 0.428050146574035 -0.079981116260155]**	**Stable Mode**
λ_3	**0.993050417160544 + 0.117689714831879i**	v_3	**[0.000000000000404 + 0.071490040308861i 0.565805752353912 + 0.000000000000000i 0.000000000001040 + 0.533886944142501i 0.557293966293566 - 0.000000000000116i**	**Oscillatory Mode**

			-0.000000000003205 - 0.233733574231927i 0.156556082678370 - 0.000000000003063i]	
λ_4	**0.993050417160544 - 0.117689714831879i**	v_4	**[0.000000000000404 - 0.071490040308861i 0.565805752353912 + 0.000000000000000i 0.000000000001040 - 0.533886944142501i 0.557293966293566 + 0.000000000000116i -0.000000000003205 + 0.233733574231927i 0.156556082678370 + 0.000000000003063i]**	**Oscillator Mode**
λ_5	**0.999999999999926 + 0.000000505577048i**	v_5	**[-0.000000000000533 - 0.000000276433744i -0.704768496602233 + 0.000000000000000i -0.000000000003664 - 0.000006532533562i -0.677482849387889 + 0.000000000000000i 0.000000000002847 - 0.000001526932367i 0.210519250752074 + 0.000000000000000i]**	**Trivial**
λ_6	**0.999999999999926 - 0.000000505577048i**	v_6	**[-0.000000000000533 + 0.000000276433744i -0.704768496602233 + 0.000000000000000i -0.000000000003664 + 0.000006532533562i -0.677482849387889 - 0.000000000000000i 0.000000000002847 + 0.000001526932367i 0.210519250752074 - 0.000000000000000i]**	**Trivial**

- As seen, the last two eigenvalues are the trivial eigenvalues as they are closest to 1. The first pair is a reciprocal pair and the second pair is a complex conjugate pair. But, the magnitude of the second pair is 1 and hence, the reciprocal is simply the complex conjugate. So, all the pairs are reciprocals.

- Stability of the orbit
  - The eigenvalues of the monodromy matrix can be used to derive conclusions about the stability of the orbit. There are 6 eigenvalues, 2 of which are real and the remaining 4 are complex. 2 of the complex conjugate can be considered trivial and hence, ignored.
  - The real eigenvalues have 1 eigenvalue that is greater than 1 (unstable mode) and 1 eigenvalue that is less than 1 (stable mode).
  - The complex eigenvalues have a magnitude of 0.999999999999970 ~ 1 and so they lie on the unit circle and hence, correspond to oscillatory modes.
  - So, there are stable, unstable, and oscillatory modes.
  - These are indicated in the last column in the above table.
- Structure of the solution space in the local neighbourhood of the periodic orbit
  - The stable mode has a positive real eigenvalue. This means that a small step around the reference point means that the orbit will remain on the same side of the reference trajectory and the final state will get closer to the reference trajectory than the initial state.
  - The unstable mode also has a positive real eigenvalue. This means that a small step around the reference point means that the orbit will remain on the same side of the reference trajectory and the final state will get farther away from the reference trajectory than the initial state.

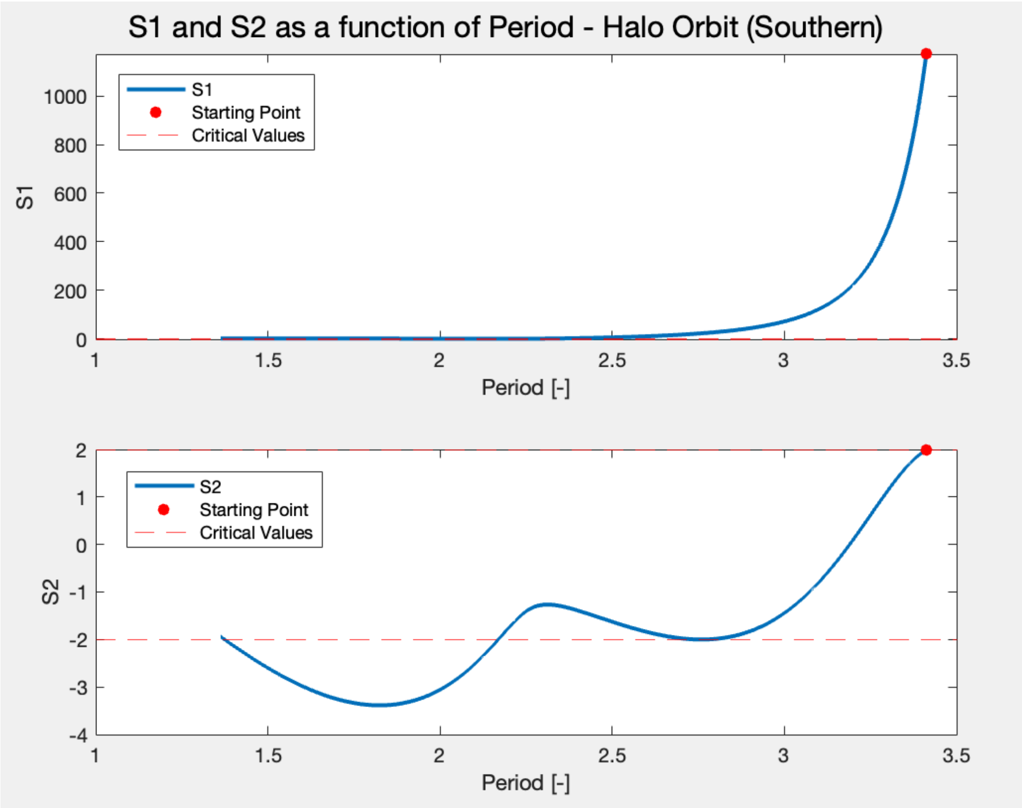
Problem 1 - Part c

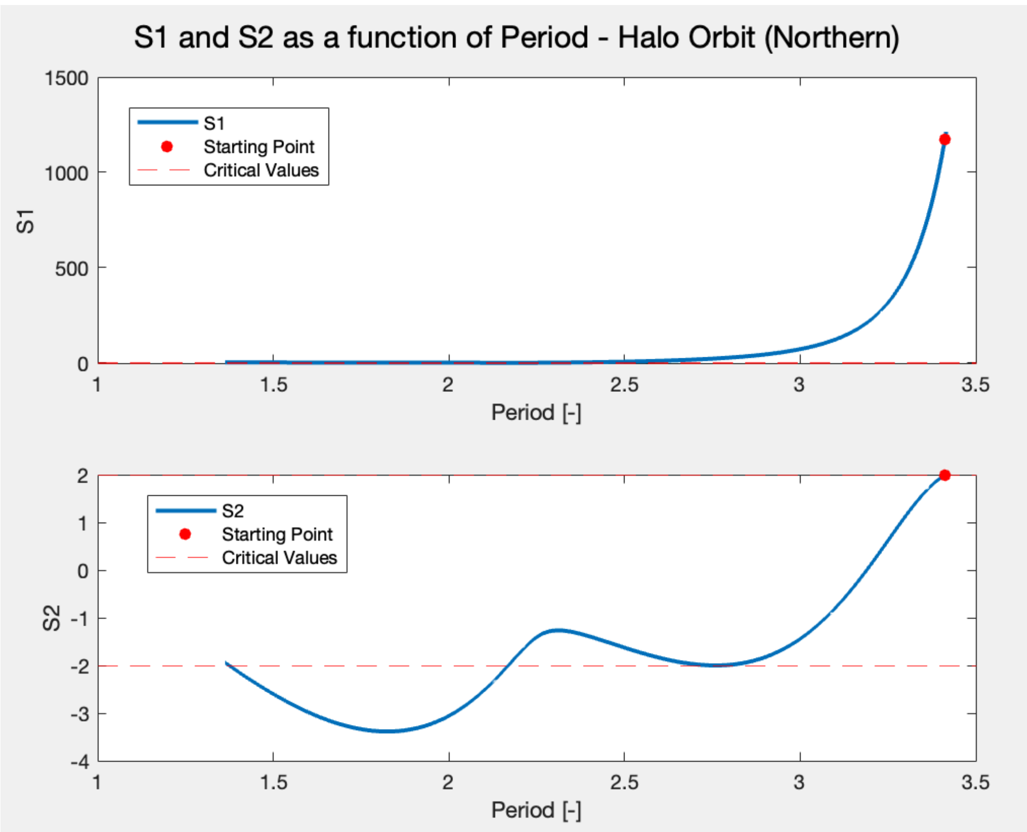
To assess the stability of the periodic orbit families, a stability index is used. This stability index is the sum of the non-trivial reciprocal eigenvalues. So, each orbit has 2 stability indices. The following process is used to arrange the eigenvalues so that the correct eigenvalues are added together:

- Loop through the eigenvalues and find reciprocal pairs.
  - These can be found by reciprocating the current eigenvalue and finding the closest eigenvalue from the remaining eigenvalue.
  - This provides 3 stability indices. All three indices are subtracted by 2 (because the trivial stability index will be 2) and the smallest of these indices will be the trivial pair and can be ignored.
- Then, both the stability indices are compared with the previous indices to assign s1 and s2.

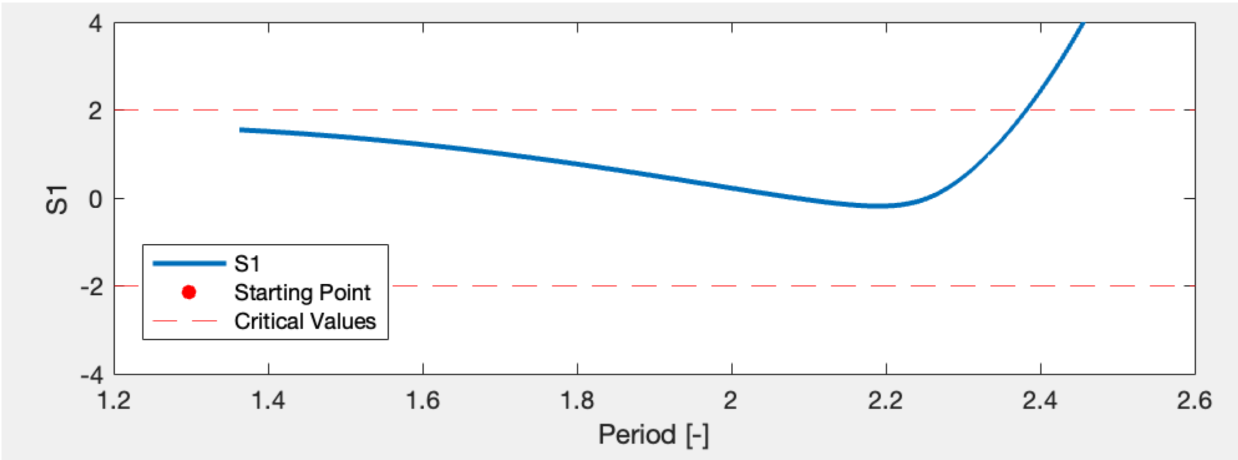
These stability indices are used to assess the stability of various orbits along the family. If the stability index of an orbit is greater than 2 or less than -2, that indicates that stable, and unstable modes are present for the orbit. If the stability index is less than 2 and greater than -2, that indicates the orbit has an oscillatory mode. When the stability index is 2 or -2, the orbit stability changes and may lead to bifurcations.

This leads to the following stability index plot as a function of the orbit period for the L2 halo orbit family:
(The critical values are +2 and -2 as discussed above)


S1 and S2 as a function of Period - Halo Orbit (Southern)
S1
Starting Point
Critical Values
S1
Period [-]
S2
Starting Point
Critical Values
S2
Period [-]

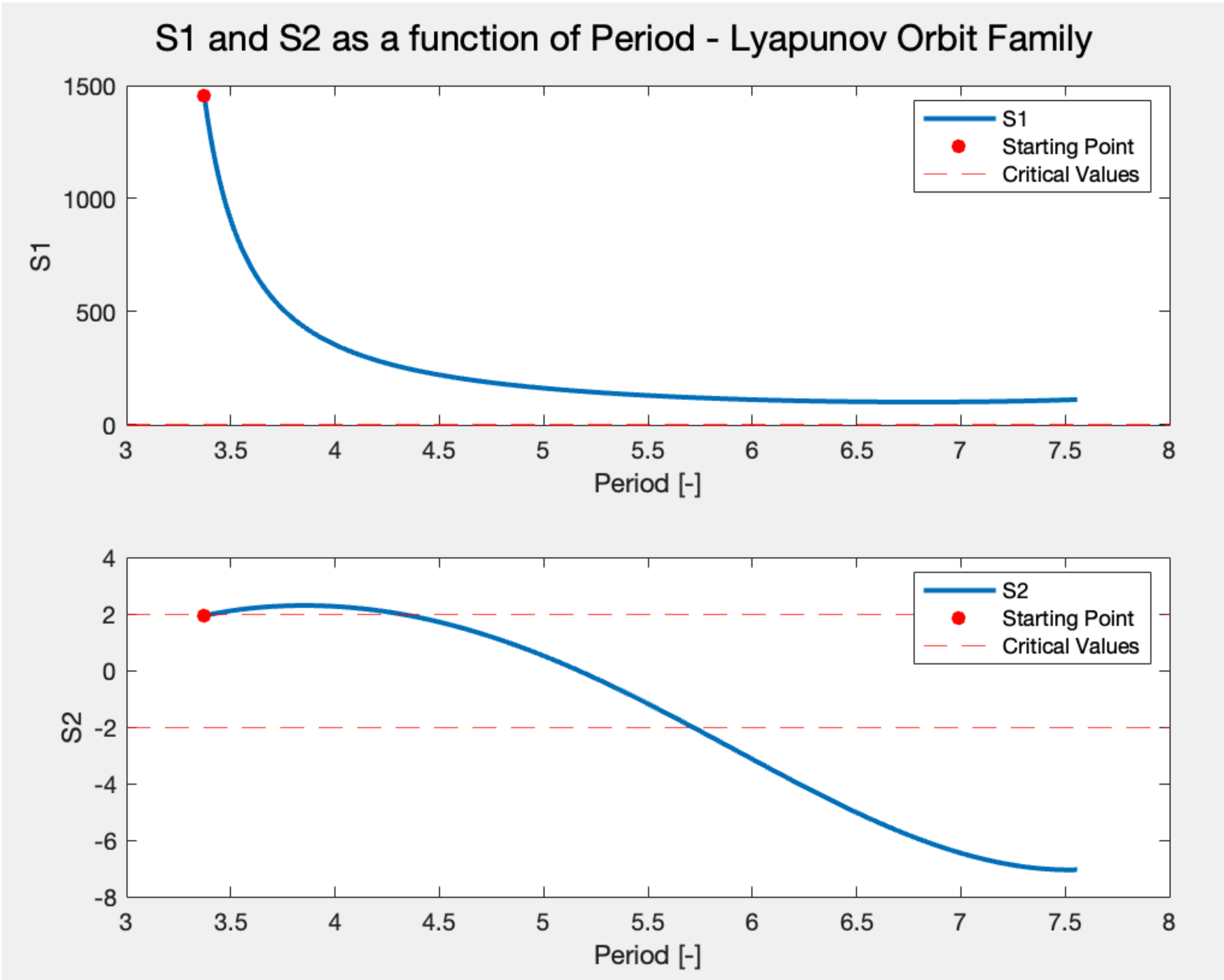

S1 and S2 as a function of Period - Halo Orbit (Northern)
S1
Starting Point
Critical Values
S1
Period [-]
S2
Starting Point
Critical Values
S2
Period [-]

As seen above the northern and southern stability indices look very similar. Here is a zoomed-in version of the S1 stability index to better help visualize:



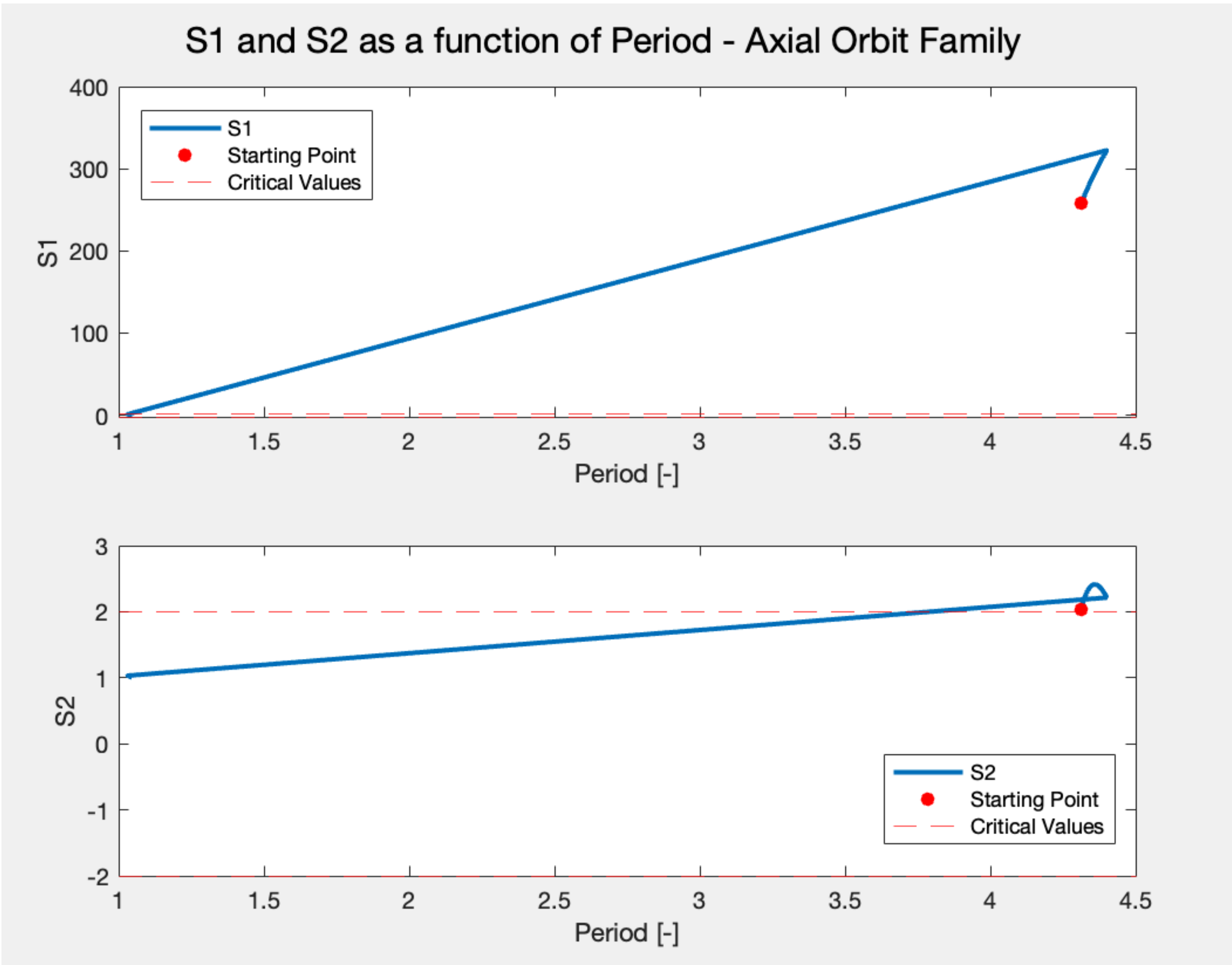
The S1 stability index passes +2, so there is a change in the stability - going from stable/unstable modes to oscillatory modes. At +2, there can also be a period multiplying bifurcation. Additionally, a period multiplying bifurcation can occur whenever the stability index is between the critical values of +/- 2. The S2 stability index crosses the critical value +2 which changes the stability of the orbit. Then, it has oscillatory modes which can lead to quasi-periodic trajectories and period-multiplying bifurcations. Then, it has a tangent bifurcation at -2 and a -2 critical value can also lead to period-doubling bifurcations. Lastly, it strays below -2 which are also stable/unstable modes (can lead to twisting orbits because s < -2) and then it terminates at -2 again (period-doubling bifurcation).

Similarly, below is stability index plot as a function of the orbit period for the L2 Lyapunov orbit family:

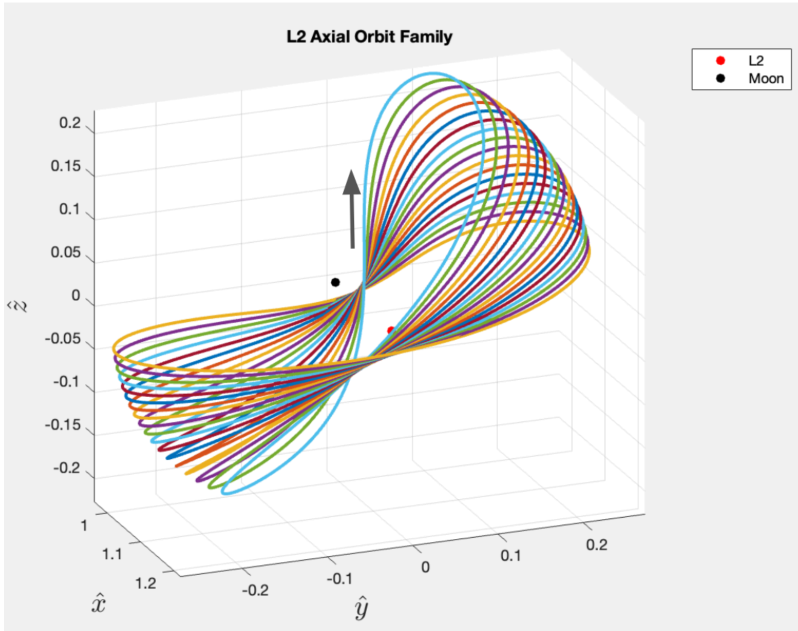


The S1 stability index is always above the critical values, and that's why it always maintains the stable/unstable modes. Since the Lyapunov orbits are planar, the S1 indices indicate in-plane modes. So, the in-plane modes are always stable/unstable. The S2 indices indicate out-of-plane modes. It can be seen that the S2 stability indices start around +2 so it's harder to infer what sort of bifurcation occurs there. But, then it has stable/unstable modes (>2) and it crosses +2 again which might lead to other period-multiplying bifurcations. Lastly, it crosses -2 which can lead to a period-doubling bifurcation. Another inference is that the out-of-plane modes may have a bifurcation with L2 Axial orbits which is partly why my continuation seems to fail while trying to compute vertical orbits.

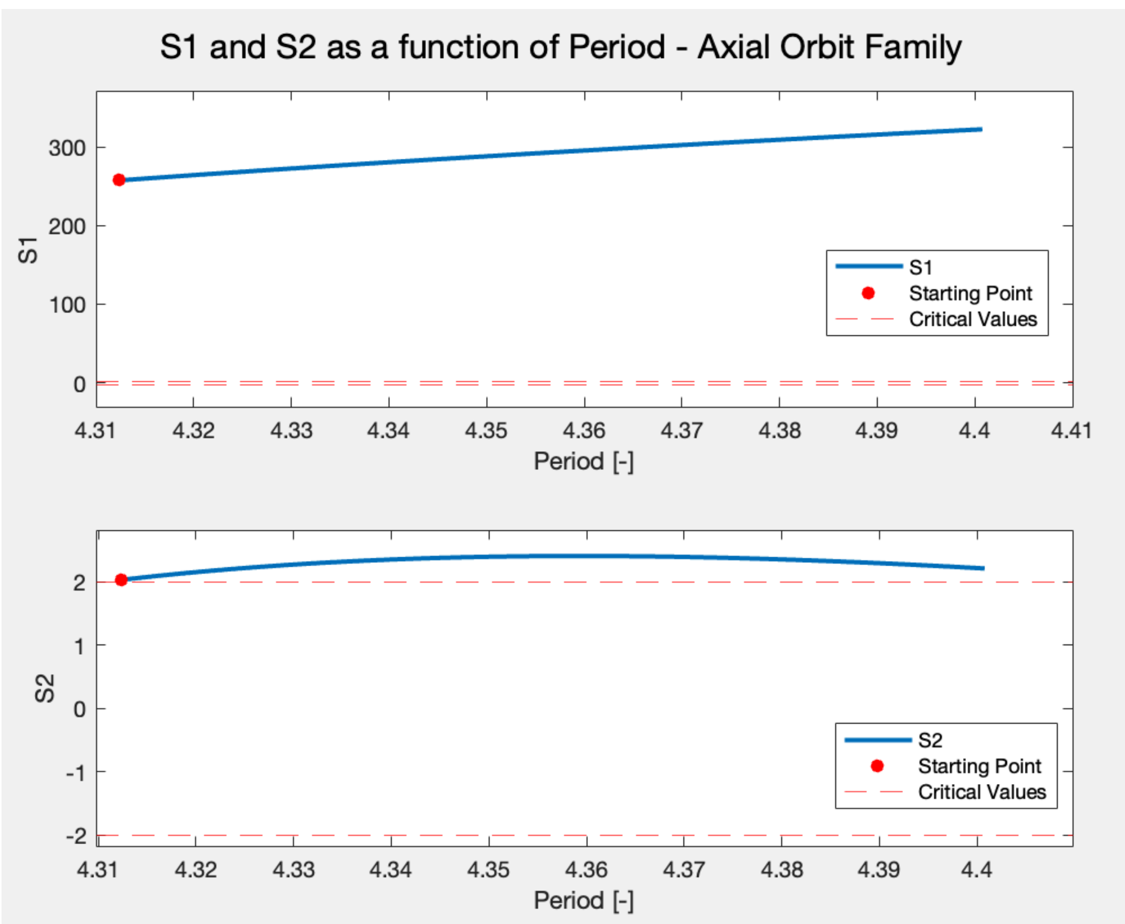
Similarly, below is stability index plot as a function of the orbit period for the L2 Axial orbit family:



Since, I see the discontinuity near vertical orbits, I don't think it is appropriate to infer stability from a discontinuous family. Instead, I will only use the first half of the family computed by a negative step size. The family looks like this:



The stability index as a function of period plot is as follows:



The stability indices seem incomplete because the family isn't complete. For the discontinuous family, S1 doesn't seem to be near the critical values and so it can be assumed that S1 has stable/unstable modes. S2 however, starts very close to the +2 critical value which can lead to period-multiplicity bifurcations. Since, we don't know what happened before the starting point, it could have been a well-known local bifurcation. However, after the starting point, the stability index is >2 which indicates stable/unstable modes.

Problem 1 - Appendix

Halo orbit monodromy matrix with higher precision.

```
halo_monodromy =

   1.0e+03 *

  Columns 1 through 4

   0.679067278824166  -0.255826911024016  -0.050309975519209   0.262763218451909
  -0.239575644730570   0.091032570326028   0.017854446563800  -0.092480144978599
  -0.027883469559164   0.010491927952806   0.003066413579085  -0.010835937870768
   1.276687359648529  -0.480079235121683  -0.094647722799498   0.494106988867356
  -0.695990181873914   0.262597432199950   0.051640512297527  -0.269699525879394
  -0.130356615927373   0.048979438740944   0.009673286101583  -0.050309975519344

  Columns 5 through 6

   0.092480144978814  -0.010835937870738
  -0.032779227915072   0.003792561008944
  -0.003792561008963   0.000252795053643
   0.174017188901118  -0.020298347541214
  -0.093927719631245   0.011179947788682
  -0.017854446563889   0.003066413579087
```

Problem 2 - Part a

The inputs to this function are time steps and state outputs (42x1) of the numerical integrator i.e. discrete time and states (42x1) of the periodic orbit. Additionally, equilibrium point position, system mass ratio, and manifold propagation time are also needed.

The following procedure is used to compute the manifolds for a period orbit:

- Firstly, an event function is newly added.
    - The event function sets 3 values based on state inputs
        - Value
            - The value that we want to be 0.
            - For the moon, we want the x value to be 1-μ, so we want $x - (1 - \mu) = 0$
            - For the earth, we want the x value to be -μ, so we want $x - \mu = 0$
        - isTerminal
            - This value is set to 1 to indicate to ODE that the integration needs to be terminated when value = 0.
        - Direction
            - This indicates which direction of the function at the terminal point. It is set to 0 to terminate the integration from either side.
- Then, a step size value is set - d
    - d is set to 50 km and normalized by the average semi-major axis of the earth-moon system (384,400 km from HW1 constants).
        - This value is provided from lectures and corroborated by the manifolds later in this problem.
- Then, a figure is instantiated and the Lyapunov orbit is plotted, along with the equilibrium point, earth, and moon.
- The STM matrix for the initial state is obtained from the input state - $\Phi(t_1 + T,\ t_1)$.
    - It is the STM at the last step of the input state.
- A for loop is instantiated to run for each step in the input state.
    - An STM from the current state to a full period in the future is required – $\Phi(t_j + T,\ t_j)$. This is obtained from the following equation:

      $$\Phi(t_j + T,\ t_j) = \Phi(t_j, t_1) * \Phi(t_1 + T,\ t_1) * \Phi(t_j, t_1)^{-1}$$

    - Eigenvalues of this STM are obtained using Matlab's eig() function.
    - Then, the trivial eigenvalues of the STM are found by subtracting each value by 1 and checking the remainder. The index of the remainder array with the minimal value is the index of the trivial eigenvalue.
    - Then, only the real eigenvalues are checked for stable and unstable modes.
        - If a real eigenvalue has a value less than 1, its index is registered as stable.
        - If a real eigenvalue has a value greater than 2, its index is registered as unstable.

- The stable and unstable eigenvalues and corresponding eigenvectors are obtained using these indices and normalized by the position states (first 3 elements of the eigenvectors).
- Then, the stable and unstable steps are calculated using the normalized eigenvectors using the following equations:

$$\bar{x}_s = \bar{x}_{PO} \pm d\bar{v}^s(\bar{x}_{PO})$$

$$\bar{x}_u = \bar{x}_{PO} \pm d\bar{v}^u(\bar{x}_{PO})$$

- These are classified as moon-bound or earth-bound based on the x-velocity of the stable and unstable vectors: $\bar{x}_s$, $\bar{x}_u$
    - If the x-velocity is positive, that manifold is moon-bound
    - If the x-velocity is negative, that manifold is earth-bound
- These stable and unstable manifolds are then numerically integrated using ODE113() and the event functions are also passed to make sure the propagation stops at the moon and the earth respectively.
- Then, these trajectories are plotted.
- This loop is repeated for each state along the periodic orbit.

The function is shown below:

Table of Contents


```
function manifolds(tout, xout, mu, l1_pos, manifold_time)
    % Script to compute stable/unstable manifolds for a periodic orbit
    % Inputs:
    % tout - discrete time steps
    % xout - 42x1 discrete state vectors
    % mu - system mass ratio
    % l1_pos - equilibrium point position
    % manifold_time - time to propagate manifold forward/backward
    %
    % Outputs:
    % Graph with stable/unstable manifolds

    % Set options for ode113()
    % Part b
    % options = odeset('RelTol', 1e-12, 'AbsTol', 1e-12, 'Events', @(t,y)
eventFn(t, y, mu));

    % Part c - ignore event function
    options = odeset('RelTol', 1e-12, 'AbsTol', 1e-12);

    a = 384400; % [kg] EM average SMA
    d = 50 / a; % [-] Unitless, normalized by a

    period = tout(end);

    p1_pos = [-mu, 0, 0];
    p2_pos = [1-mu, 0, 0];

    figure()
    plot(xout(:,1), xout(:,2), 'black', 'LineWidth', 3)
    hold on
    scatter(l1_pos(1), l1_pos(2), 'filled', 'red')
    scatter(p1_pos(1), p1_pos(2), 'filled', 'blue')
    scatter(p2_pos(1), p2_pos(2), 'filled', ' black')

    % Compute STM - phi(t1+T, t1)
    phi_t1T_t1 = reshape(xout(end,7:42), [6,6])';

    % Begin for loop
    for i = 1:10:length(tout)

        % Compute STM - phi(tj+T, tj)
        phi_tj_t1 = reshape(xout(i, 7:42), [6,6])';
        phi_tjT_tj = phi_tj_t1 * phi_t1T_t1 * inv(phi_tj_t1);

        % Get evals, evecs
        [V, D] = eig(phi_tjT_tj);
```

```
% Get evals as an array
for j = 1:6
    evals(j) = D(j,j);
end

% Subtract evals by 1 and get 2 minimum indices. These are trivial
% indices
evals_minus_1 = evals - 1;
[min_evals, trivial_index] = mink(evals_minus_1, 2);

% If eval is real and not trivial, assign stable and unstable
% indices
for j = 1:6
    if (isreal(evals(j)) && isnotin(trivial_index, j))
        if evals(j) < 1
            stable_index = j;
        elseif evals(j) > 1
            unstable_index = j;
        end
    end
end

% Get stable/unstable evec and normalize eigenvector by 1st 3 terms
stable_eval = D(stable_index, stable_index);
stable_evec = V(:, stable_index);
stable_pos_norm = norm(stable_evec(1:3));
stable_evec = stable_evec/stable_pos_norm;
% stable_evec(4:6) = -stable_evec(4:6);
unstable_eval = D(unstable_index, unstable_index);
unstable_evec = V(:, unstable_index);
unstable_pos_norm = norm(unstable_evec(1:3));
unstable_evec = unstable_evec/unstable_pos_norm;

% Step into manifold
x_manifold_s_p = xout(i,1:6)' + d * stable_evec;
x_manifold_s_n = xout(i,1:6)' - d * stable_evec;

% ONLY FOR L1
% If x-velocity is positive, moon-bound
% If x-velocity if negative, earth-bound
if (x_manifold_s_p(4) > 0)
    moon_stable = x_manifold_s_p;
    earth_stable = x_manifold_s_n;
else
    moon_stable = x_manifold_s_n;
    earth_stable = x_manifold_s_p;
end
% Repeat for unstable manifolds
x_manifold_u_p = xout(i,1:6)' + d * unstable_evec;
x_manifold_u_n = xout(i,1:6)' - d * unstable_evec;
if (x_manifold_u_p(4) > 0)
    moon_unstable = x_manifold_u_p;
    earth_unstable = x_manifold_u_n;
```

```
        else
            moon_unstable = x_manifold_u_n;
            earth_unstable = x_manifold_u_p;
        end

        % Propagate using the event functions
        [moon_stable_t, moon_stable_x] = ode113(@(t, state)CR3BP(state, mu),
[0, -manifold_time], moon_stable, options);
        [moon_unstable_t, moon_unstable_x] = ode113(@(t, state)CR3BP(state,
mu), [0, manifold_time], moon_unstable, options);
        [earth_stable_t, earth_stable_x] = ode113(@(t, state)CR3BP(state,
mu), [0, -manifold_time], earth_stable, options);
        [earth_unstable_t, earth_unstable_x] = ode113(@(t, state)CR3BP(state,
mu), [0, manifold_time], earth_unstable, options);

        plot(moon_stable_x(:,1), moon_stable_x(:,2), 'blue')
        plot(earth_stable_x(:,1), earth_stable_x(:,2), 'blue')
        plot(moon_unstable_x(:,1), moon_unstable_x(:,2), 'red')
        plot(earth_unstable_x(:,1), earth_unstable_x(:,2), 'red')

    end
    hold off
    legend("Lyapunov Orbit", "L1", "Earth", "Moon")
    grid on
    axis equal
    xlabel('$$\hat{x}$$','Interpreter','Latex', 'FontSize',18)
    ylabel('$$\hat{y}$$','Interpreter','Latex', 'FontSize',18)
end
```

```
Not enough input arguments.

Error in manifolds (line 23)
    period = tout(end);
```

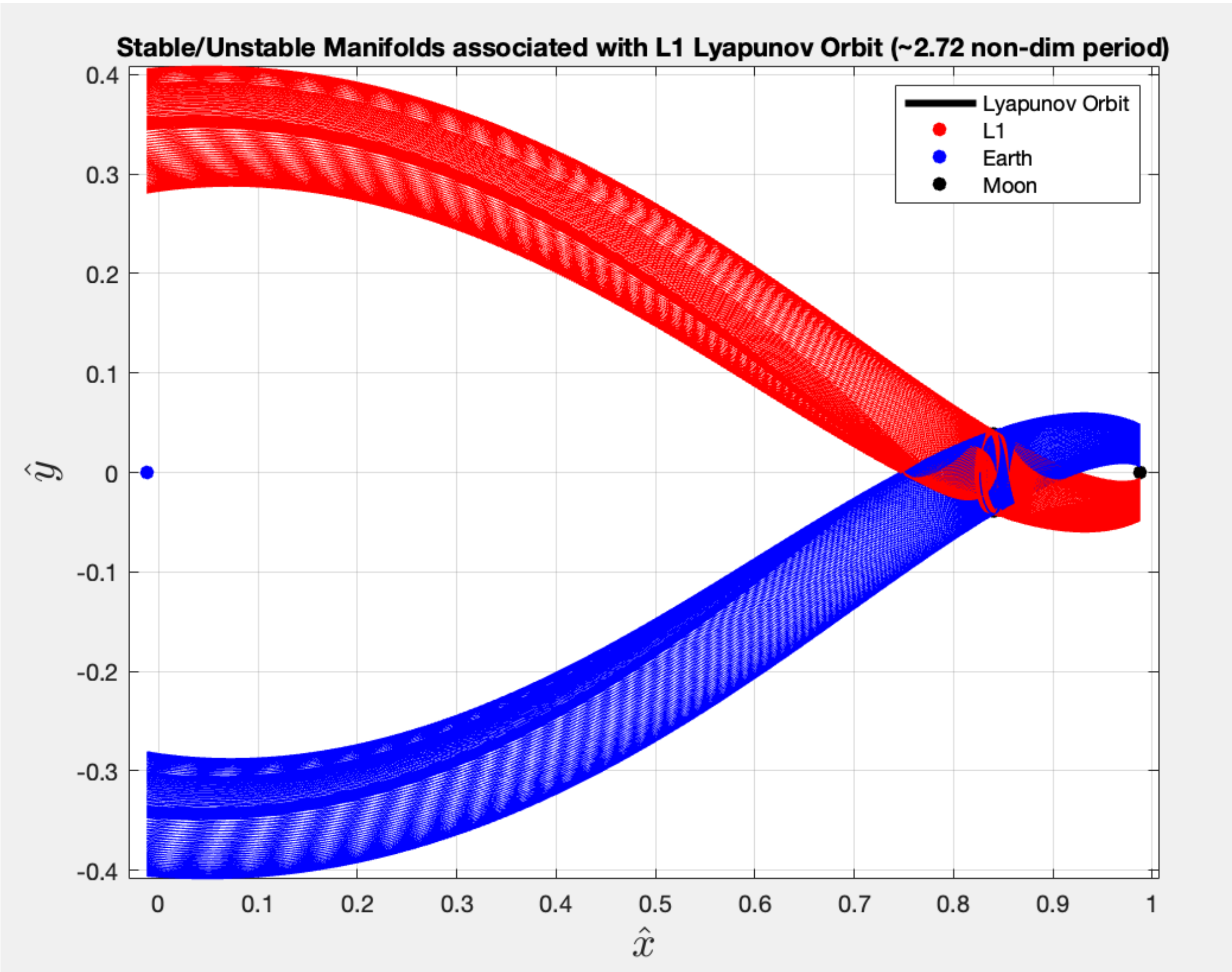
Published with MATLAB® R2024a

Problem 2 - Part b

Initial state vector for ~2.72 non-dimensional time:

$$\bar{x}_0 = [0.826400827951781,\ 0,\ 0,\ 0.000000000000003,\ 0.095740069326045,\ 0]^T$$

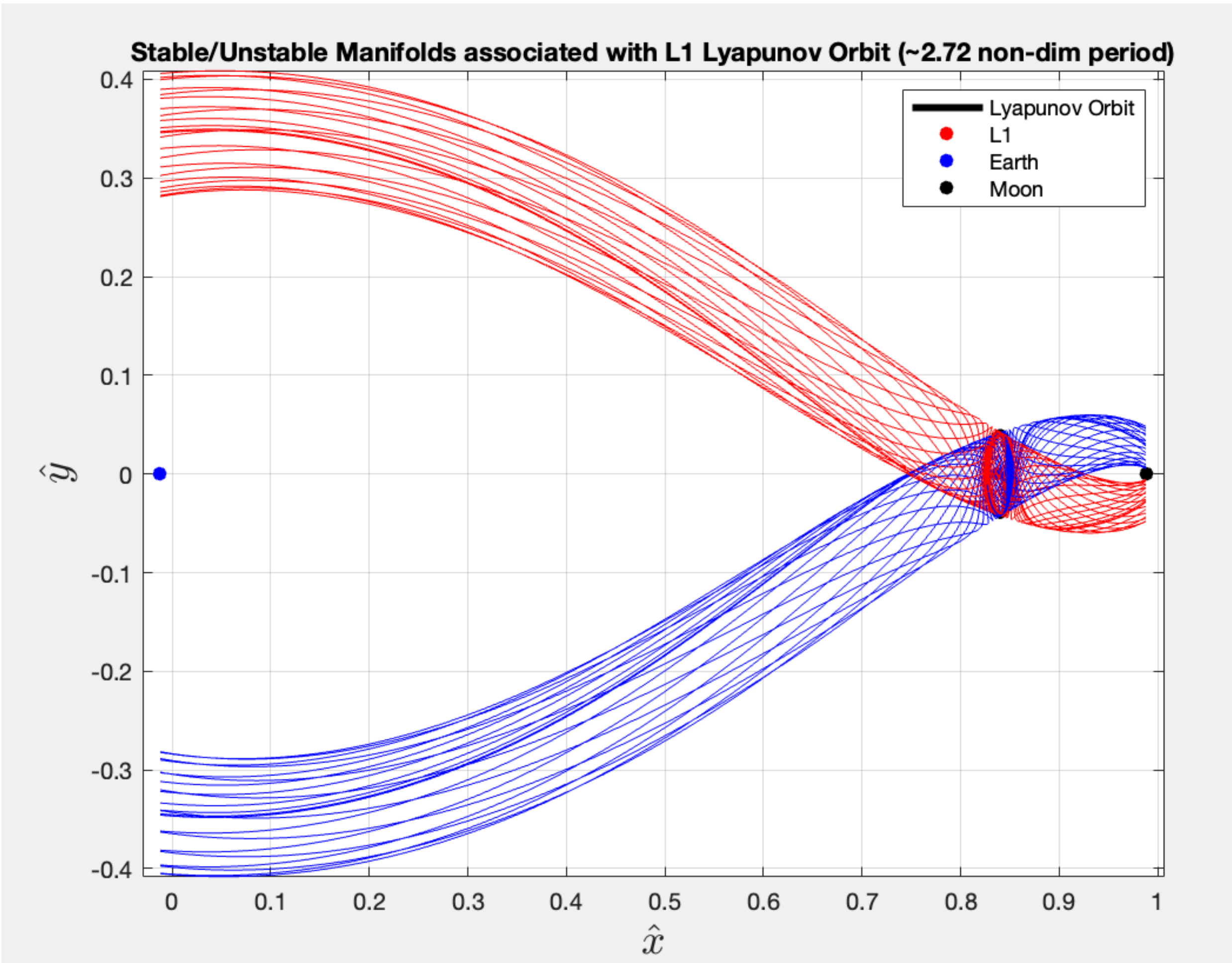
T = 2.720282553661705 [-]



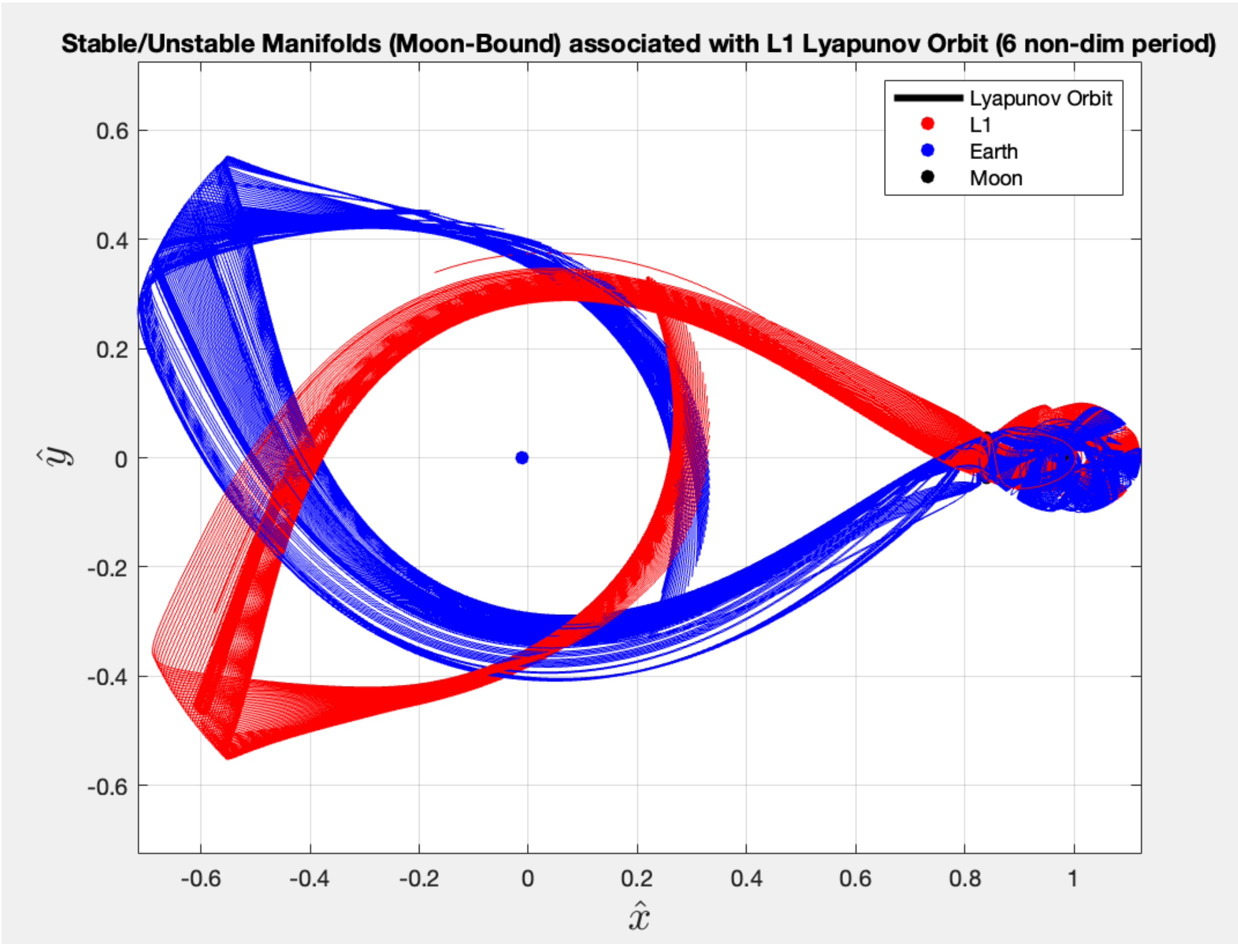
Characteristics of the computed manifolds:

- The stable and unstable manifolds are symmetric over the y axis.
- The stable and unstable manifolds intersect transversally at the L1 Lyapunov periodic orbit.
- The manifolds follow the manifolds for the L1 equilibrium points as seen in the lecture.
- These manifolds are also planar (z and $\dot{z}$) are always 0.

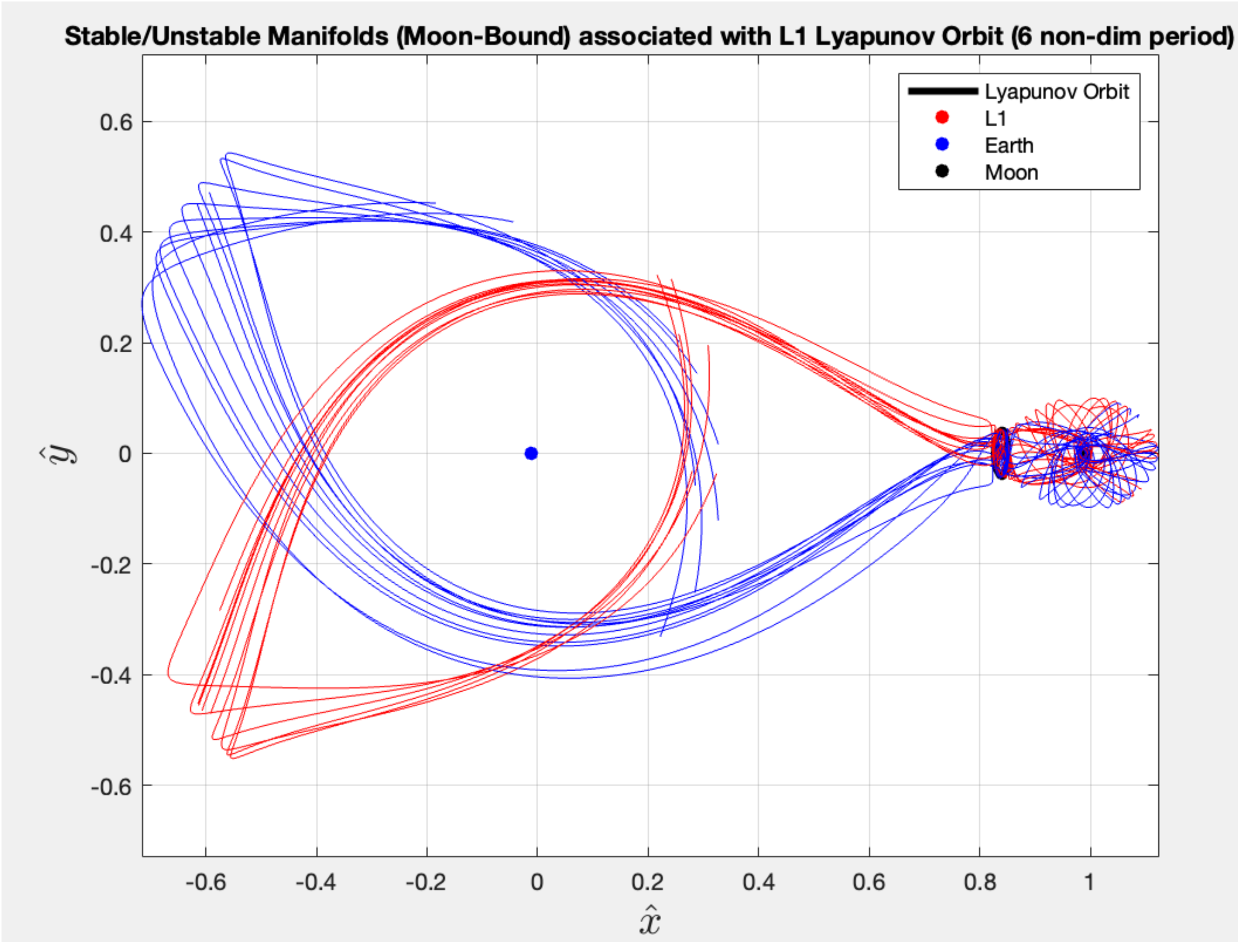
This plot is very dense. Just for visualization, next plot shows the stable/unstable manifolds for a larger discrete step size over the periodic orbit:



Problem 2 - Part c



Same as last time, let's increase the step size here to help visualize this better.

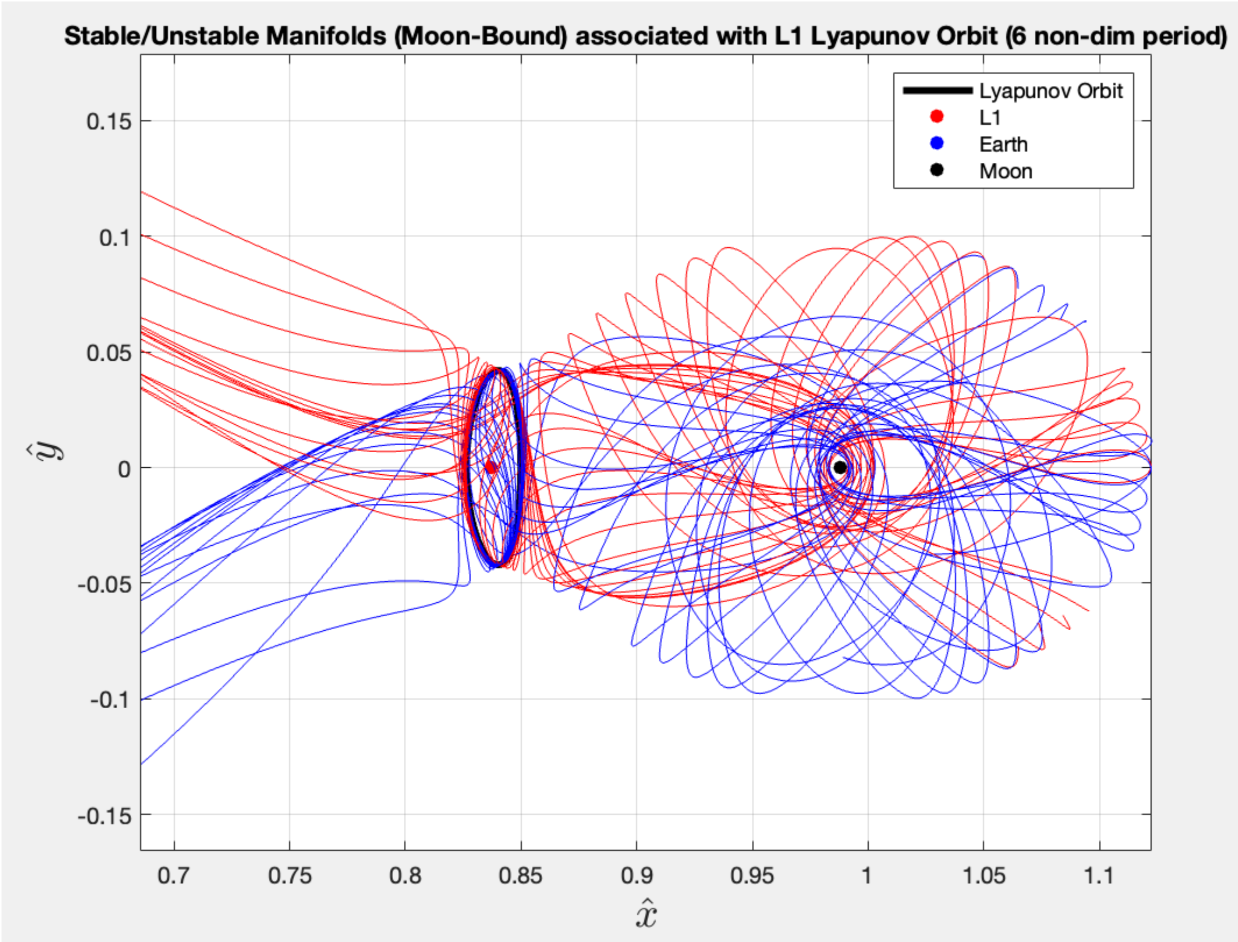


Description of the manifolds

- This manifold looks very similar to part b for some time which validates the fact that the only difference between this plot and part b is that this is propagated for a longer time period.
- There is very complex behavior around the moon.
    - One of the reasons for such complex behavior is that the P2 body (moon) is close and the dynamics are very sensitive near that region.
- The manifolds wrap around the moon or start with an initial positive x-velocity and then move towards the earth but remain within the system.
- This reminds me very much of the zero-velocity curves where the natural trajectory stays within the system because the L2 gateway is closed and keeps bouncing within near the earth-moon system.

Complexity of visualization/analysis for this more complex set of trajectories

- One measure to reduce the complexity of visualization is to increase the step size so as to plot fewer stable/unstable manifolds.
- It is also really difficult to visualize the dynamics near the moon. Here is a zoomed-in version:



Stable/Unstable Manifolds (Moon-Bound) associated with L1 Lyapunov Orbit (6 non-dim period)
Lyapunov Orbit
L1
Earth
Moon
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
0.7
0.75
0.8
0.85
0.9
0.95
1
1.05
1.1
$\hat{x}$
$\hat{y}$

Table of Contents


```
clear; clc; close all;

% ASEN 6060 - HW 4, Problem 1
% Spring 2025
% Jash Bhalavat
```

Constants

```
G = 6.67408 * 10^-11; % m3/(kgs2)
G = G / (10^9); % km3/(kgs2)

% Earth
mu_earth = 398600.435507; % km3/s2
a_earth = 149598023; % km
e_earth = 0.016708617;
mass_earth = mu_earth / G; % kg

% Moon
mu_moon = 4902.800118; % km3/s2
a_moon = 384400; % km
e_moon = 0.05490;
mass_moon = mu_moon / G; % kg

% Earth-Moon system
mass_ratio_em = mass_moon / (mass_earth + mass_moon);
m_star_em = mass_earth + mass_moon;
l_star_em = a_moon;
t_star_em = sqrt(l_star_em^3/(G * m_star_em));
mu = mass_ratio_em;
```

Part a - i)

```
% Get L2 Point
% Earth Moon system equilibrium points
```

```
[em_eq_pts, em_eq_validity] = all_eq_points(mu);

% Only looking at L2 eq point planar oscillatory modes
l2_pos = [em_eq_pts(2,:), 0];

l2_in_plane_modes = in_plane_modes(mu, l2_pos);

oscillatory_eval = l2_in_plane_modes(3);
uxx_l2 = u_xx(mu, l2_pos);
uxy_l2 = u_xy(mu, l2_pos);
uyy_l2 = u_yy(mu, l2_pos);
U_star_XX = [uxx_l2, uxy_l2; uxy_l2, uyy_l2];
Omega = [0 2; -2 0];

A2D = [zeros(2), eye(2); U_star_XX, Omega];

[V, D] = eig(A2D);

oscillatory_evec = real(V(:,3));

oscillatory_pos_mag = norm([oscillatory_evec(1), oscillatory_evec(2)]);

pos_mag_req = 0.0001;

oscillatory_mag_factor = pos_mag_req / oscillatory_pos_mag;

oscillatory_ic = oscillatory_evec .* oscillatory_mag_factor;

% Time is one period
t = linspace(0, 2*pi/imag(oscillatory_eval), 1000);

xi_0 = oscillatory_ic(1);
xi_dot_0 = oscillatory_ic(3);
eta_0 = oscillatory_ic(2);
eta_dot_0 = oscillatory_ic(4);
x0 = [l2_pos(1) + xi_0; l2_pos(2) + eta_0; 0; xi_dot_0; eta_dot_0; 0];

TOL = 5e-14;

% Set options for ode113
options = odeset('RelTol', TOL, 'AbsTol', TOL);
[tout, xout] = ode113(@(t, state)CR3BP(state, mu), [0 t(end)], x0, options);

V0 = [x0; t(end)];

V_soln = gen_3d_periodic_orbit_single_shooting(V0, mu, false);

[tout_corrected, xout_corrected] = ode113(@(t, state)CR3BP(state, mu), [0,
V_soln(end)], V_soln(1:6), options);

% V_family_a1 = test_psal_cont(V_soln, mu);
% save("V_family_a1.mat", "V_family_a1")
load("V_family_a1.mat")
```

```
p1_pos = [-mu, 0, 0];
p2_pos = [1-mu, 0, 0];

figure()
scatter(l2_pos(1), l2_pos(2), 'filled', 'red')
hold on
scatter(p1_pos(1), p1_pos(2), 'filled', 'blue')
scatter(p2_pos(1), p2_pos(2), 'filled', 'black')

for i = 1:50:size(V_family_a1, 2)
    [tout, xout] = ode113(@(t,state)CR3BP(state, mu), [0, V_family_a1(7,i)],
V_family_a1(1:6,i), options);
    plot(xout(:,1), xout(:,2), 'LineWidth',2)
end
hold off
legend("L2", "Earth", "Moon")
xlabel('$$\hat{x}$$','Interpreter','Latex', 'FontSize',18)
ylabel('$$\hat{y}$$','Interpreter','Latex', 'FontSize',18)
grid on
axis equal
title("L2 Lyapunov Orbit Family")
```

Part a - ii)

```
x0 = [1.180462, 0, -0.0209998, 0, -0.158363, 0]';
T = 3.411921;
V0 = [x0; T];

V_soln = gen_3d_periodic_orbit_single_shooting(V0, mu, true);

[tout_corrected, xout_corrected] = ode113(@(t, state)CR3BP(state, mu), [0,
V_soln(end)], V_soln(1:6), options);

% V_family_a2 = test_psal_cont(V_soln, mu);
% save("V_family_a2.mat", "V_family_a2")
load("V_family_a2.mat")
V_family_a2_south = V_family_a2;
load("V_family_a2_pos_delta_s.mat")
V_family_a2_north = V_family_a2;

figure()
p1 = scatter3(l2_pos(1), l2_pos(2), l2_pos(3), 'filled', 'red');
hold on
p2 = scatter3(p2_pos(1), p2_pos(2), p2_pos(3), 'filled', 'black');

for i = 1:250:size(V_family_a2_south, 2)
    [tout, xout] = ode113(@(t,state)CR3BP(state, mu), [0,
V_family_a2_south(7,i)], V_family_a2_south(1:6,i), options);
    p3 = plot3(xout(:,1), xout(:,2), xout(:,3), 'LineWidth',2, 'Color',
'red');
end

for i = 1:250:size(V_family_a2_north, 2)
```

```
    [tout, xout] = ode113(@(t,state)CR3BP(state, mu), [0,
V_family_a2_north(7,i)], V_family_a2_north(1:6,i), options);
    p4 = plot3(xout(:,1), xout(:,2), xout(:,3), 'LineWidth',2, 'Color',
'blue');
end
hold off
p = [p1, p2, p3, p4];
legend(p, "L2", "Moon", "Southern Halo", "Northern Halo")
xlabel('$$\hat{x}$$','Interpreter','Latex', 'FontSize',18)
ylabel('$$\hat{y}$$','Interpreter','Latex', 'FontSize',18)
zlabel('$$\hat{z}$$','Interpreter','Latex', 'FontSize',18)
grid on
axis equal
title("L2 Halo Orbit Family")
```

Part a - iii)

```
x0 = [1.0301513, 0,0, 0, 0.7030025,0.1552945]';
T = 4.312367;
V0 = [x0; T];

V_soln = gen_3d_periodic_orbit_single_shooting(V0, mu, false);

[tout_corrected, xout_corrected] = ode113(@(t, state)CR3BP(state, mu), [0,
V_soln(end)], V_soln(1:6), options);

V_soln = [xout_corrected(284,:)'; V_soln(end)];

delta_s_neg = -1e-3;
delta_s_pos = 1e-3;
% V_family_a3_neg_ds = pseudo_arc_length_continuation(V_soln, mu,
delta_s_neg);
% V_family_a3_pos_ds = pseudo_arc_length_continuation(V_soln, mu,
delta_s_pos);
load("V_family_a3_neg_ds.mat")
load("V_family_a3_pos_ds.mat")

figure()
scatter3(l2_pos(1), l2_pos(2), l2_pos(3), 'filled', 'red')
hold on
scatter3(p2_pos(1), p2_pos(2), p2_pos(3), 'filled', 'black')

for i = 1:50:size(V_family_a3_neg_ds, 2)
    [tout, xout] = ode113(@(t,state)CR3BP(state, mu), [0,
V_family_a3_neg_ds(7,i)], V_family_a3_neg_ds(1:6,i), options);
    plot3(xout(:,1), xout(:,2), xout(:,3), 'LineWidth',2)
end

for i = 1:50:size(V_family_a3_pos_ds, 2)
    [tout, xout] = ode113(@(t,state)CR3BP(state, mu), [0,
V_family_a3_pos_ds(7,i)], V_family_a3_pos_ds(1:6,i), options);
    plot3(xout(:,1), xout(:,2), xout(:,3), 'LineWidth',2)
end
```

```
hold off
legend("L2", "Moon")
xlabel('$$\hat{x}$$','Interpreter','Latex', 'FontSize',18)
ylabel('$$\hat{y}$$','Interpreter','Latex', 'FontSize',18)
zlabel('$$\hat{z}$$','Interpreter','Latex', 'FontSize',18)
grid on
axis equal
title("L2 Axial Orbit Family")
% save("V_family_a3.mat", "V_family_a3")
```

Part b - L2 Halo first orbit

```
identity_row = reshape(eye(6), [36,1]);
[tout, xout] = ode113(@(t,state)CR3BP_full(state, mu), [0, V_family_a2(7,1)],
[V_family_a2(1:6,1); identity_row], options);

halo_monodromy = reshape(xout(end,7:42), [6,6])';
[V_halo, D_halo] = eig(halo_monodromy);

[s1, s2, period] = stability_index(V_family_a2_north, mu, identity_row,
options);
figure()
subplot(2,1,1)
plot(period, s1, 'LineWidth', 2)
hold on
scatter(period(1), s1(1), 'filled', 'red')
yline(2, '--r')
yline(-2, '--r')
xlabel("Period [-]")
ylabel("S1")
legend("S1", "Starting Point", "Critical Values")
hold off
subplot(2,1,2)
plot(period, s2, 'LineWidth', 2)
hold on
scatter(period(1), s2(1), 'filled', 'red')
yline(2, '--r')
yline(-2, '--r')
xlabel("Period [-]")
ylabel("S2")
legend("S2", "Starting Point", "Critical Values")
hold off
sgtitle("S1 and S2 as a function of Period - Halo Orbit (Northern)")
```

Part c - Halo Orbit Family

```
[s1, s2, period] = stability_index(V_family_a2_south, mu, identity_row,
options);
figure()
subplot(2,1,1)
plot(period, s1, 'LineWidth', 2)
hold on
scatter(period(1), s1(1), 'filled', 'red')
```

```
yline(2, '--r')
yline(-2, '--r')
xlabel("Period [-]")
ylabel("S1")
legend("S1", "Starting Point", "Critical Values")
hold off
subplot(2,1,2)
plot(period, s2, 'LineWidth', 2)
hold on
scatter(period(1), s2(1), 'filled', 'red')
yline(2, '--r')
yline(-2, '--r')
xlabel("Period [-]")
ylabel("S2")
legend("S2", "Starting Point", "Critical Values")
hold off
sgtitle("S1 and S2 as a function of Period - Halo Orbit (Southern)")
```

Part c - L2 Lyapunov orbit family

```
[tout, xout] = ode113(@(t,state)CR3BP_full(state, mu), [0, V_family_a1(7,1)],
[V_family_a1(1:6,1); identity_row], options);

lyapunov_monodromy = reshape(xout(end,7:42), [6,6])';
[V_lyapunov, D_lyapunov] = eig(lyapunov_monodromy);

% [s1, s2, period] = stability_index(V_family_a1, mu, identity_row, options);
figure()
subplot(2,1,1)
plot(period, s1, 'LineWidth', 2)
hold on
scatter(period(1), s1(1), 'filled', 'red')
yline(2, '--r')
yline(-2, '--r')
xlabel("Period [-]")
ylabel("S1")
legend("S1", "Starting Point", "Critical Values")
hold off
subplot(2,1,2)
plot(period, s2, 'LineWidth', 2)
hold on
scatter(period(1), s2(1), 'filled', 'red')
yline(2, '--r')
yline(-2, '--r')
xlabel("Period [-]")
ylabel("S2")
legend("S2", "Starting Point", "Critical Values")
hold off
sgtitle("S1 and S2 as a function of Period - Lyapunov Orbit Family")
```

Part c - L2 Axial orbit family

```
V_family_a3 = [V_family_a3_neg_ds, flip(V_family_a3_pos_ds)];
% V_family_a3 = V_family_a3_neg_ds;

[tout, xout] = ode113(@(t,state)CR3BP_full(state, mu), [0, V_family_a3(7,1)],
[V_family_a3(1:6,1); identity_row], options);

[s1, s2, period] = stability_index(V_family_a3, mu, identity_row, options);
figure()
subplot(2,1,1)
plot(period, s1, 'LineWidth', 2)
hold on
scatter(period(1), s1(1), 'filled', 'red')
yline(2, '--r')
yline(-2, '--r')
xlabel("Period [-]")
ylabel("S1")
legend("S1", "Starting Point", "Critical Values")
hold off
subplot(2,1,2)
plot(period, s2, 'LineWidth', 2)
hold on
scatter(period(1), s2(1), 'filled', 'red')
yline(2, '--r')
yline(-2, '--r')
xlabel("Period [-]")
ylabel("S2")
legend("S2", "Starting Point", "Critical Values")
hold off
sgtitle("S1 and S2 as a function of Period - Axial Orbit Family")
```

Problem 2

```
% ASEN 6060 - HW 4, Problem 2
% Spring 2025
% Jash Bhalavat
```

Part b

```
mu = mass_ratio_em;

% Earth Moon system equilibrium points
[em_eq_pts, em_eq_validity] = all_eq_points(mu);

% Only looking at L1 eq point planar oscillatory modes
l1_pos = [em_eq_pts(1,:), 0];

l1_in_plane_modes = in_plane_modes(mu, l1_pos);

oscillatory_eval = l1_in_plane_modes(3);
uxx_l1 = u_xx(mu, l1_pos);
```

```
uxy_l1 = u_xy(mu, l1_pos);
uyy_l1 = u_yy(mu, l1_pos);
U_star_XX = [uxx_l1, uxy_l1; uxy_l1, uyy_l1];

Omega = [0 2; -2 0];

A2D = [zeros(2), eye(2); U_star_XX, Omega];

[V, D] = eig(A2D);

oscillatory_evec = real(V(:,3));

oscillatory_pos_mag = norm([oscillatory_evec(1), oscillatory_evec(2)]);

pos_mag_req = 0.0001;

oscillatory_mag_factor = pos_mag_req / oscillatory_pos_mag;

oscillatory_ic = oscillatory_evec .* oscillatory_mag_factor;

% Time is one period
t = linspace(0, 2*pi/imag(oscillatory_eval), 1000);

xi_0 = oscillatory_ic(1);
xi_dot_0 = oscillatory_ic(3);
eta_0 = oscillatory_ic(2);
eta_dot_0 = oscillatory_ic(4);
x0 = [l1_pos(1) + xi_0; l1_pos(2) + eta_0; 0; xi_dot_0; eta_dot_0; 0];

% Set options for ode113
options = odeset('RelTol', 1e-12, 'AbsTol', 1e-12);
[tout, xout] = ode113(@(t, state)CR3BP(state, mu), [0 t(end)], x0, options);

V0 = [x0; t(end)];

V_soln = gen_3d_periodic_orbit_single_shooting(V0, mu, false);

[tout_corrected, xout_corrected] = ode113(@(t, state)CR3BP_full(state, mu),
[0, V_soln(end)], [V_soln(1:6); reshape(eye(6), [36,1])], options);

% V_family = pseudo_arc_length_continuation(V_soln, mu);
load("V_family_L1_Lyapunov.mat");
% save("V_family_L1_Lyapunov.mat", V_family);

p1_pos = [-mu, 0, 0];
p2_pos = [1-mu, 0, 0];

figure()
scatter(l1_pos(1), l1_pos(2), 'filled', 'red')
hold on
scatter(p1_pos(1), p1_pos(2), 'filled', 'blue')
scatter(p2_pos(1), p2_pos(2), 'filled', 'black')

for i = 1:50:size(V_family, 2)
```

```
    [tout, xout] = ode113(@(t,state)CR3BP(state, mu), [0, V_family(7,i)],
V_family(1:6,i), options);
    plot3(xout(:,1), xout(:,2), xout(:,3), 'LineWidth',2)
end
hold off
legend("L2", "Earth", "Moon")
xlabel('$$\hat{x}$$','Interpreter','Latex', 'FontSize',18)
ylabel('$$\hat{y}$$','Interpreter','Latex', 'FontSize',18)
grid on
axis equal
title("L1 Lyapunov Orbit Family")


% 2.72 nondim time is 102 index
[tout, xout] = ode113(@(t,state)CR3BP_full(state, mu), [0, V_family(7,102)],
[V_family(1:6,102); reshape(eye(6), [36,1])], options);

manifolds(tout, xout, mu, l1_pos, 10);
title("Stable/Unstable Manifolds associated with L1 Lyapunov Orbit (~2.72 non-
dim period)")
```

Part c

```
manifolds(tout, xout, mu, l1_pos, 6);
title("Stable/Unstable Manifolds (Moon-Bound) associated with L1 Lyapunov
Orbit (6 non-dim period)")
```

Functions

```
function [s1, s2, period] = stability_index(V_family, mu, identity_row,
options)
    counter = 1;

    for i = 1:size(V_family, 2)
        [tout, xout] = ode113(@(t,state)CR3BP_full(state, mu), [0,
V_family(7,i)], [V_family(1:6,i); identity_row], options);
        monodromy_i = reshape(xout(end,7:42), [6,6])';
        [evecs, evals] = eig(monodromy_i);

        % Pass to function
        [s1_i, s2_i] = arrange_evals(evals, evecs);

        if (counter==1)
            s1(counter) = s1_i;
            s2(counter) = s2_i;
        else
            diff1 = abs(s1_i - s1(counter-1));
            diff2 = abs(s1_i - s2(counter-1));
            if diff1 < diff2
                s1(counter) = s1_i;
                s2(counter) = s2_i;
            else
                s1(counter) = s2_i;
```

```
                s2(counter) = s1_i;
            end
        end

        period(counter) = V_family(7,i);
        counter = counter + 1;
    end
end

function manifolds(tout, xout, mu, l1_pos, manifold_time)
    % Script to compute stable/unstable manifolds for a periodic orbit
    % Inputs:
    % tout - discrete time steps
    % xout - 42x1 discrete state vectors
    % mu - system mass ratio
    % l1_pos - equilibrium point position
    % manifold_time - time to propagate manifold forward/backward
    %
    % Outputs:
    % Graph with stable/unstable manifolds

    % Set options for ode113()
    % Part b
    % options = odeset('RelTol', 1e-12, 'AbsTol', 1e-12, 'Events', @(t,y)
eventFn(t, y, mu));

    % Part c - ignore event function
    options = odeset('RelTol', 1e-12, 'AbsTol', 1e-12);

    a = 384400; % [kg] EM average SMA
    d = 50 / a; % [-] Unitless, normalized by a

    period = tout(end);

    p1_pos = [-mu, 0, 0];
    p2_pos = [1-mu, 0, 0];

    figure()
    plot(xout(:,1), xout(:,2), 'black', 'LineWidth', 3)
    hold on
    scatter(l1_pos(1), l1_pos(2), 'filled', 'red')
    scatter(p1_pos(1), p1_pos(2), 'filled', 'blue')
    scatter(p2_pos(1), p2_pos(2), 'filled', ' black')

    % Compute STM - phi(t1+T, t1)
    phi_t1T_t1 = reshape(xout(end,7:42), [6,6])';

    % Begin for loop
    for i = 1:10:length(tout)

        % Compute STM - phi(tj+T, tj)
        phi_tj_t1 = reshape(xout(i, 7:42), [6,6])';
        phi_tjT_tj = phi_tj_t1 * phi_t1T_t1 * inv(phi_tj_t1);
```

```
% Get evals, evecs
[V, D] = eig(phi_tjT_tj);

% Get evals as an array
for j = 1:6
    evals(j) = D(j,j);
end

% Subtract evals by 1 and get 2 minimum indices. These are trivial
% indices
evals_minus_1 = evals - 1;
[min_evals, trivial_index] = mink(evals_minus_1, 2);

% If eval is real and not trivial, assign stable and unstable
% indices
for j = 1:6
    if (isreal(evals(j)) && isnotin(trivial_index, j))
        if evals(j) < 1
            stable_index = j;
        elseif evals(j) > 1
            unstable_index = j;
        end
    end
end

% Get stable/unstable evec and normalize eigenvector by 1st 3 terms
stable_eval = D(stable_index, stable_index);
stable_evec = V(:, stable_index);
stable_pos_norm = norm(stable_evec(1:3));
stable_evec = stable_evec/stable_pos_norm;
% stable_evec(4:6) = -stable_evec(4:6);
unstable_eval = D(unstable_index, unstable_index);
unstable_evec = V(:, unstable_index);
unstable_pos_norm = norm(unstable_evec(1:3));
unstable_evec = unstable_evec/unstable_pos_norm;

% Step into manifold
x_manifold_s_p = xout(i,1:6)' + d * stable_evec;
x_manifold_s_n = xout(i,1:6)' - d * stable_evec;

% ONLY FOR L1
% If x-velocity is positive, moon-bound
% If x-velocity if negative, earth-bound
if (x_manifold_s_p(4) > 0)
    moon_stable = x_manifold_s_p;
    earth_stable = x_manifold_s_n;
else
    moon_stable = x_manifold_s_n;
    earth_stable = x_manifold_s_p;
end
% Repeat for unstable manifolds
x_manifold_u_p = xout(i,1:6)' + d * unstable_evec;
x_manifold_u_n = xout(i,1:6)' - d * unstable_evec;
if (x_manifold_u_p(4) > 0)
```

```
            moon_unstable = x_manifold_u_p;
            earth_unstable = x_manifold_u_n;
        else
            moon_unstable = x_manifold_u_n;
            earth_unstable = x_manifold_u_p;
        end

        % Propagate using the event functions
        [moon_stable_t, moon_stable_x] = ode113(@(t, state)CR3BP(state, mu),
[0, -manifold_time], moon_stable, options);
        [moon_unstable_t, moon_unstable_x] = ode113(@(t, state)CR3BP(state,
mu), [0, manifold_time], moon_unstable, options);
        [earth_stable_t, earth_stable_x] = ode113(@(t, state)CR3BP(state,
mu), [0, -manifold_time], earth_stable, options);
        [earth_unstable_t, earth_unstable_x] = ode113(@(t, state)CR3BP(state,
mu), [0, manifold_time], earth_unstable, options);

        plot(moon_stable_x(:,1), moon_stable_x(:,2), 'blue')
        plot(earth_stable_x(:,1), earth_stable_x(:,2), 'blue')
        plot(moon_unstable_x(:,1), moon_unstable_x(:,2), 'red')
        plot(earth_unstable_x(:,1), earth_unstable_x(:,2), 'red')

    end
    hold off
    legend("Lyapunov Orbit", "L1", "Earth", "Moon")
    grid on
    axis equal
    xlabel('$$\hat{x}$$','Interpreter','Latex', 'FontSize',18)
    ylabel('$$\hat{y}$$','Interpreter','Latex', 'FontSize',18)
end

function [value,isterminal,direction] = eventFn(t,y, mu)
    value = [1-mu-y(1), y(1)-(-mu)];
    isterminal = [1, 1]; % Halt integration when value is 0
    direction = [0, 0]; % When zero is approached from either side
end


function out = isnotin(array, val)
    out = true;
    for el = 1:length(array)
        if val == array(el)
            out = false;
        end
    end
end

function [s1, s2] = arrange_evals(evals_mat, evecs)
    for j = 1:6
        evals(j) = evals_mat(j,j);
    end
    % Get first element and its reciprocal and remove from array
    eval1 = evals(1);
    over_eval1 = 1/eval1;
```

```
    evals(1) = [];

    diff = evals - over_eval1;
    [min_diff, pair1_index] = min(abs(diff));
    pair1 = eval1 + evals(pair1_index);

    evals(pair1_index) = [];

    eval2 = evals(1);
    over_eval2 = 1/eval2;
    evals(1) = [];

    diff = evals - over_eval2;
    [min_diff, pair2_index] = min(abs(diff));
    pair2 = eval2 + evals(pair2_index);

    evals(pair2_index) = [];

    pair3 = evals(1) + evals(2);

    pairs = [pair1, pair2, pair3];
    pairs_minus_2 = pairs - 2;
    [min_pair, min_pair_ind] = min(abs(pairs_minus_2));

    pairs(min_pair_ind) = [];
    s1 = pairs(1);
    s2 = pairs(2);

end
```

Published with MATLAB® R2024a