ASEN 6060 - HW 1
Spring 2025
Jash Bhalavat

Problem 1
Part a

ASEN 6060
Spring 2025
Jash Bhalavat

(1)

$\boxed{\text{HW1}}$

Problem 1 →

Given: $GM_e = 398600.435507 \ km^3/s^2$, $a_e = 149598023 \ km$, $e_e = 0.016708617$

$GM_m = 4902.800118 \ km^3/s^2$, $a_m = 384{,}400 \ km$, $e_m = 0.05490$

$GM_{sun} = 132\,712\,440\,041.279419 \ km^3/s^2$

$G = 6.67408 \times 10^{-11} \ \frac{m^3}{kg\,s^2} = 6.67408 \times 10^{-20} \ \frac{km^3}{kg\,s^2}$

Assumptions: CR3BP

a) $\tilde{m}_e = \frac{GM_e}{G} = 5.9724 \times 10^{24} \ kg$ $\qquad \tilde{M}_{sun} = \frac{GM_{sun}}{G} = 1.9885 \times 10^{30} \ kg$

$\tilde{m}_m = \frac{GM_M}{G} = 7.3460 \times 10^{22} \ kg$

(P_1 = earth, P_2 = moon)

Earth-moon System → ($\tilde{m}_1 = \tilde{m}_e$, $\tilde{m}_2 = \tilde{m}_m$)

$$\boxed{\begin{aligned} \mu_{EM} &= \frac{\tilde{m}_2}{\tilde{m}_1 + \tilde{m}_2} = 0.0121505844 \\ m^*_{EM} &= \tilde{m}_1 + \tilde{m}_2 = 6.0458255763 \times 10^{24} \ kg \\ l^*_{EM} &= |\bar{R}_{12}| = a_m = 384{,}400 \ km \\ t^*_{EM} &= \sqrt{\frac{l^{*3}_{EM}}{\tilde{G} m^*_{EM}}} = 3.7519026189 \times 10^{5} \ s \end{aligned}}$$

P_1 = sun, P_2 = earth

Sun-Earth system → ($\tilde{m}_1 = \tilde{m}_{sun}$, $\tilde{m}_2 = \tilde{m}_e$)

$$\boxed{\begin{aligned} \mu_{SE} &= \frac{\tilde{m}_2}{\tilde{m}_1 + \tilde{m}_2} = 3.0034805945 \times 10^{-6} \\ m^*_{SE} &= \tilde{m}_1 + \tilde{m}_2 = 1.9884813883 \times 10^{30} \ kg \\ l^*_{SE} &= |\bar{R}_{12}| = a_e = 149\,598\,023 \ km \\ t^*_{SE} &= \sqrt{\frac{l^{*3}_{SE}}{\tilde{G} m^*_{SE}}} = 5.0226430182 \times 10^{6} \ s \end{aligned}}$$

Problem 1
Part b

CR3BP assumptions	Earth-Moon System	Sun-Earth System
Model only gravitational interactions between 3 bodies	This assumption is valid because only the s/c and the Earth and the Moon are modeled.	This assumption is valid because only the s/c and the Earth and the Sun are modeled. But, it must be noted that there are several bodies between the earth and the sun that will influence this system when higher fidelity modeling is conducted.
Model each body with constant mass and the same gravity field as a point mass	Earth and Moon have constant masses, but the gravity fields of Earth and Moon do not align with a point mass. So, this assumption is not fully valid.	Earth has constant mass, but the gravity field of Earth does not align with a point mass. So, this assumption is not fully valid. Also, due to the nuclear reactions in the Sun's core, its mass is constantly decreasing albeit slowly.
Mass of P3 << Masses of each P1, P2	No spacecraft until now has been close to the mass of the Earth or the Moon so this assumption is valid. If this dynamical model is used for a larger body, then this assumption may be invalid.	No spacecraft until now has been close to the mass of the Earth or the Sun so this assumption is valid. If this dynamical model is used for a larger body, then this assumption may be invalid.
P3 does not influence paths of P1 and P2	No spacecraft will be heavy enough to influence the Earth or the Moon so this assumption	No spacecraft will be heavy enough to influence the Earth or the Sun so this assumption

	is valid. If this dynamical model is used for a larger body, then this assumption may be invalid.	is valid. If this dynamical model is used for a larger body, then this assumption may be invalid.
P1 and P2 follow circular orbits	Earth and Moon don't follow circular orbits. So, this assumption is not fully valid.	The Earth doesn't follow a circular orbit around the Sun, so this assumption is not fully valid.

As seen above, a lot of the assumptions are invalid (when compared rigorously), which leads to the conclusion that the CR3BP is not the exact dynamical model for the Earth-Moon or Sun-Earth systems. But, it is fairly reasonable to start with this model because most of the assumptions are met for “first-pass” approximations.

Problem 2
Part a

Matlab's ODE45 function is used to numerically integrate the circular restricted 3-body problem (CR3BP). ODE45 is versatile and Matlab recommends trying ODE45 as the first method to numerically solve differential equations. Hence, ODE45 was chosen.

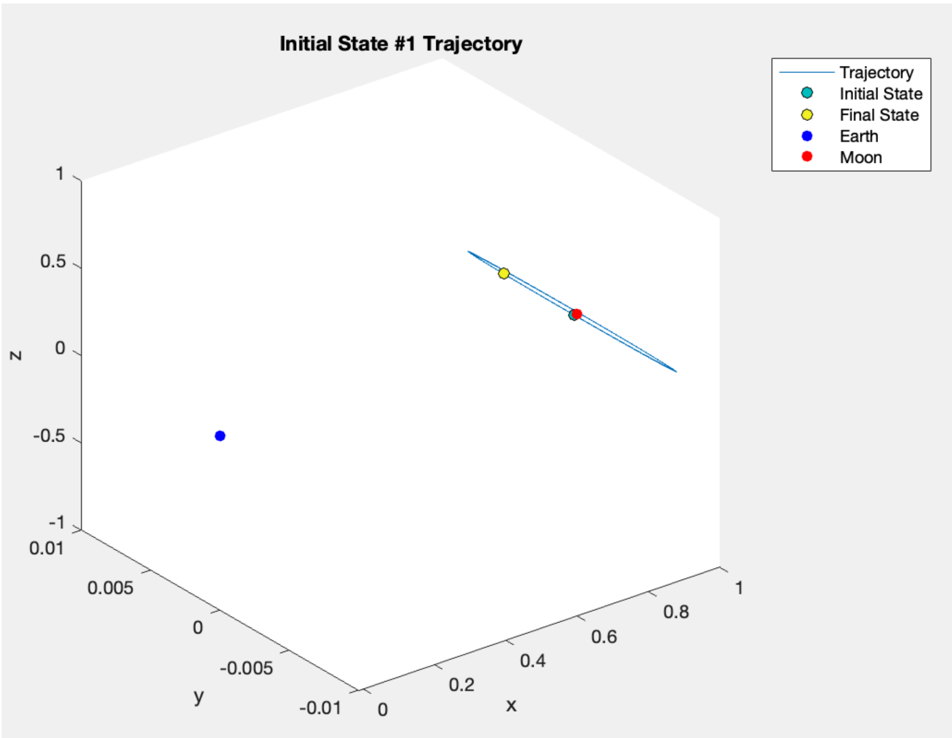
ODE45 allows setting relative and absolute tolerances. For both those tolerances, 1e-12 is used. The use of such a low tolerance can be justified because the CR3BP is a chaotic system and small changes in initial states can lead to big deviations in the output. Additionally, the cost of computing power has decreased which poses no overhead while modeling the CR3BP.

The screenshot below shows a script to numerically integrate the non-dimensional equations of motion for the CR3BP - where state0 is the initial state, int_time is the non-dimensionals integration time, and mu is the mass ratio.

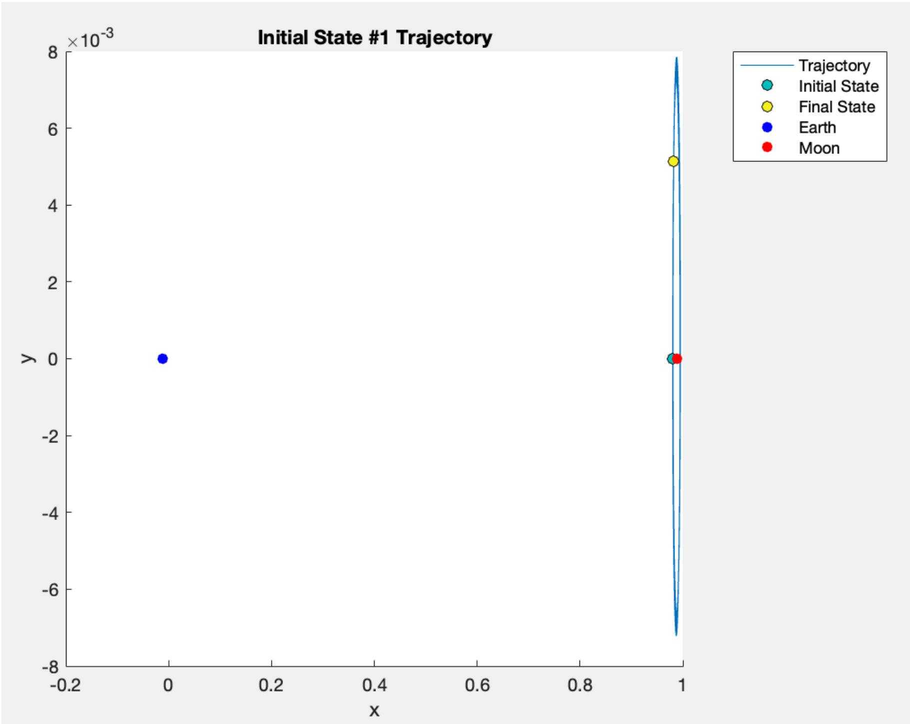
```
% Set options for ODE45
options = odeset('RelTol', 1e-12, 'AbsTol', 1e-12);

% Call ODE45 function
[tout, xout] = ode45(@(t, state)CR3BP(state, mu), [0 int_time], state0, options);

function state_dot = CR3BP(state, mu)
    % Circular Restricted 3 Body Problem non-dimensional EOMs
    x = state(1);
    y = state(2);
    z = state(3);
    xdot = state(4);
    ydot = state(5);
    zdot = state(6);

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

    state_dot(1, 1) = xdot;
    state_dot(2, 1) = ydot;
    state_dot(3, 1) = zdot;

    state_dot(4, 1) = 2*ydot + x - (1 - mu)*(x + mu)/(r1^3) - mu * (x - 1 + mu)/(r2^3);
    state_dot(5, 1) = -2*xdot + y - (1 - mu)*y/(r1^3) - mu*y/(r2^3);
    state_dot(6, 1) = - (1 - mu)*z/(r1^3) - mu*z/(r2^3);
end
```

Part b - i

This trajectory is orbiting the moon and is highly elliptical and seems like a periodic orbit around the moon. In the given time, it doesn't seem to escape the moon.

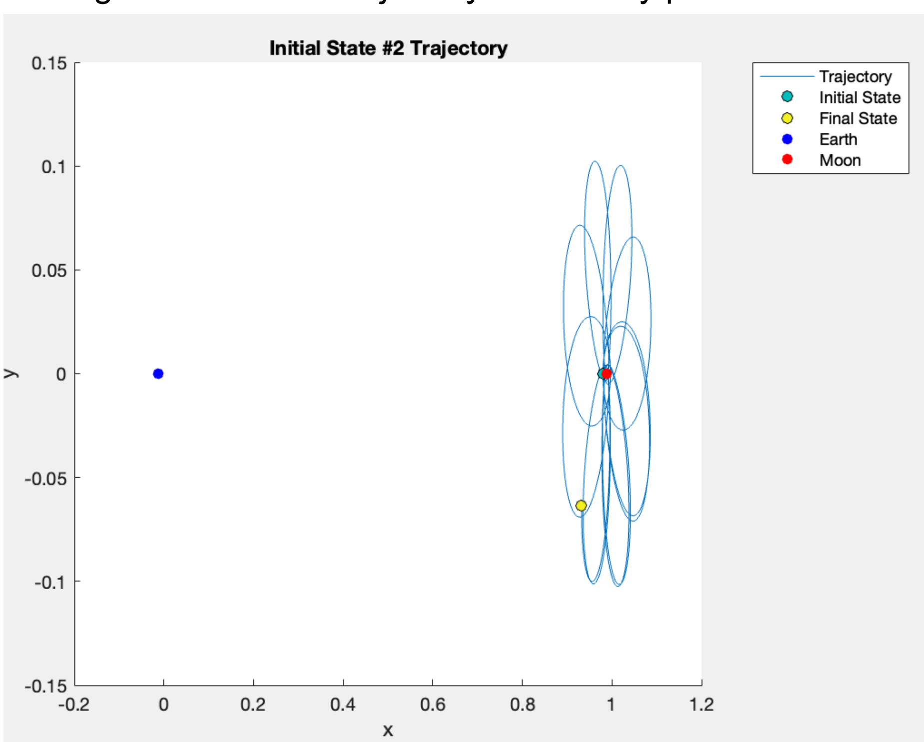


The trajectory is in the XY plane, so a top down view is shown below:



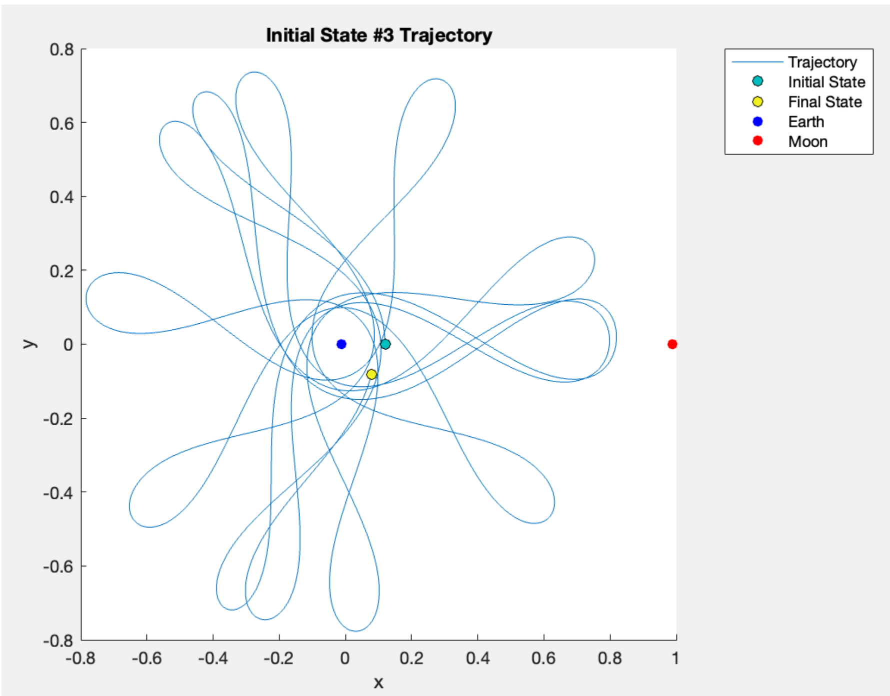
Part b - ii

This trajectory also seems periodic around the moon and doesn't leave the moon in the given time. The trajectory is in the xy plane:



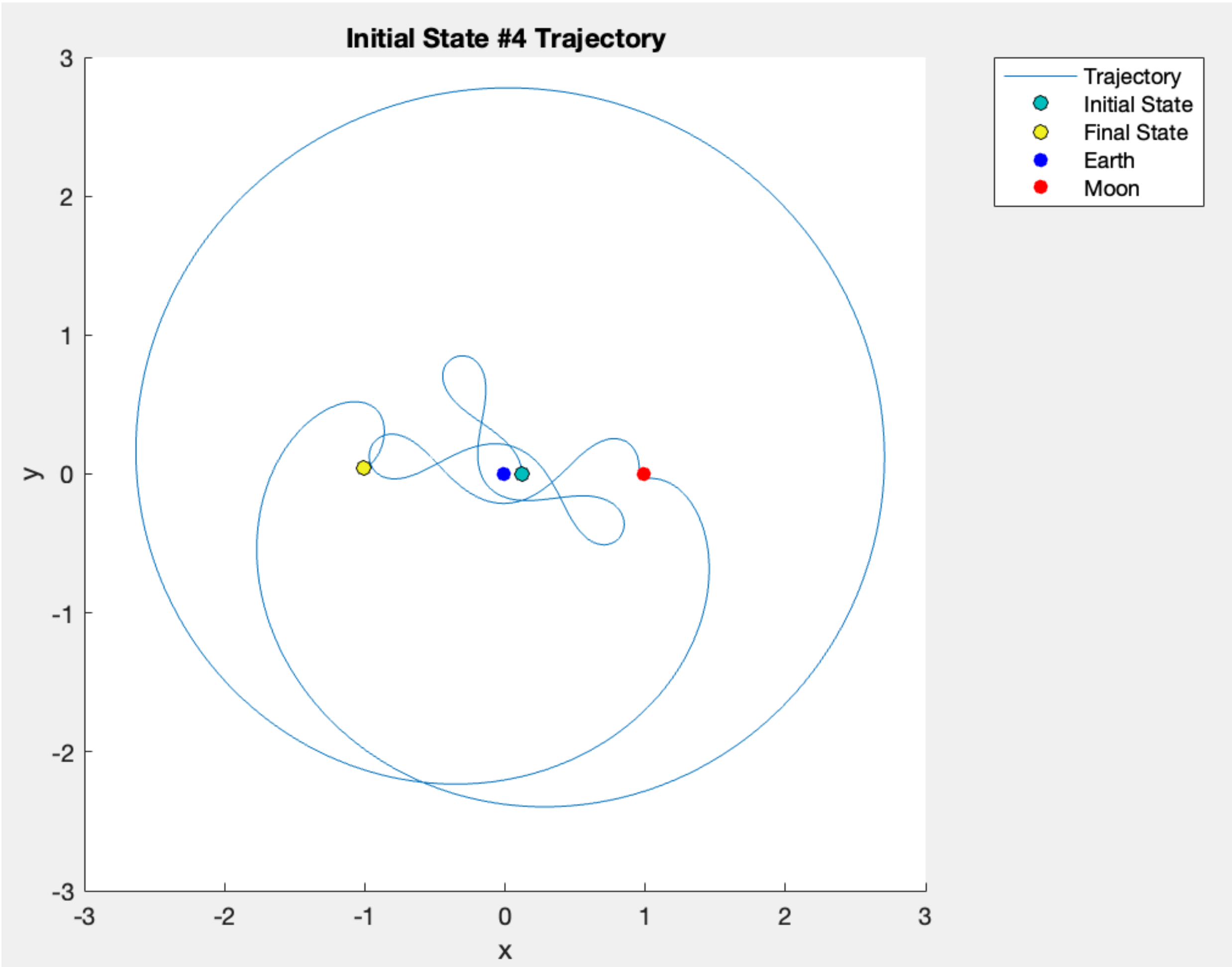
Part b - iii

The trajectory is closer to a quasi-periodic trajectory around the earth. The trajectory seems to bounce around near the earth in the given time. The trajectory is in the xy plane:



Part b - iv

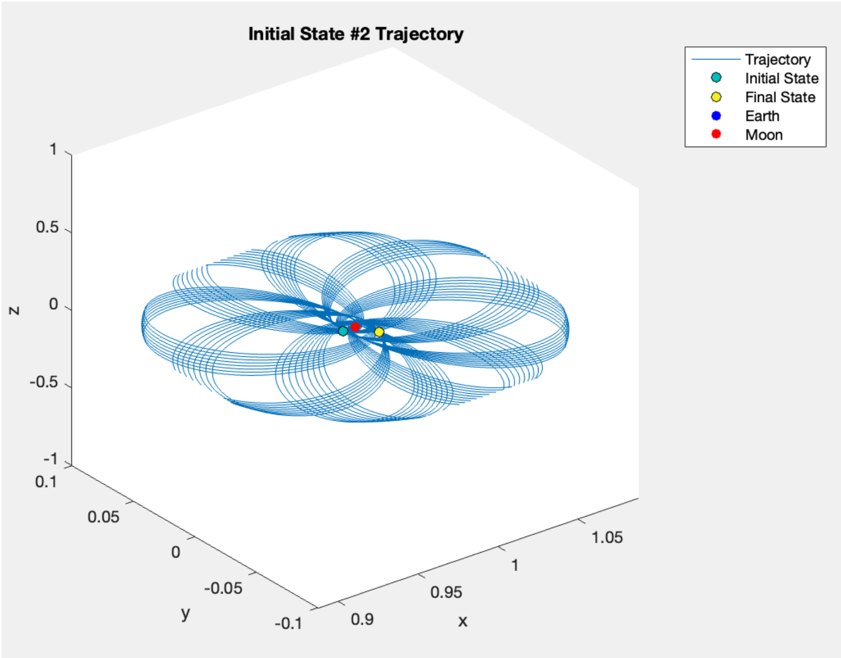
This trajectory is not periodic and starts out close to the earth and follows a similar path as part iii, but then quickly deviates to the moon and then orbits the earth-moon system once in the given time. The trajectory is in the xy plane:



Part c

Trajectory i - Increasing the non-dimensional time only shows overlap in the already existing trajectory. Hence, this trajectory is periodic. Since periodic orbits are a solution to the CR3BP EOMs, it can be concluded that this is likely a true solution.

Trajectory ii - Increasing the non-dimensional time shows the orbit shifting over time as such:



But, this doesn’t represent any true solution of the CR3BP (stable/unstable manifolds, quasi-periodic/periodic trajectory, equilibrium points, etc.). So, it can be argued that this is not likely a true solution.

Trajectory iii - A probable reason for the orbit to not escape the earth is because the jacobi constant is equal to a jacobi constant at L1 which leads to the L1 gateway being closed. That’s why the spacecraft keeps bouncing within the interior region. Hence, this orbit is likely a true solution to the EOMs of a CR3BP.

Trajectory iv - The difference between trajectory iii and iv is that trajectory iv escapes the interior region after bouncing around for a while and then it gets back closer to the earth. A probable reason for this is that the jacobi constant is between a jacobi constant at L3 and L4/L5. Since, this is one of the true solutions of the EOMs of a CR3BP, this trajectory is likely a true solution as well.

Problem 2
Part d

d) Given: Initial state $\vec{x} = [0.12, 0, 0, 0, 3.45, 0]$, $t = 25$, Earth-Moon System

Assumptions: CR3BP

Soln: $l^* = a_{moon} = 384,400$ km

$t^* = 3.7519026189 \times 10^5$ s

} from Problem 1

$\tilde{x} = x l^* = 46128$ km

$\tilde{y} = y l^* = 0$ km, $\tilde{z} = z l^* = 0$ km

$\tilde{x}' = \frac{d}{dt}\left(\frac{d\tilde{x}}{d\tau}\right) = \frac{d(xl^*)}{d(tt^*)} = \frac{l^*}{t^*} \cdot \frac{dx}{dt} = 0 \ \frac{km}{s}$

$\tilde{y}' = \frac{l^*}{t^*}\frac{dy}{dt} = 3.5346866235$ km/s

$\tilde{z}' = \frac{l^*}{t^*}\frac{dz}{dt} = 0$ km/s

Initial state

$\boxed{\tilde{x} = [46128 \text{ km}, 0 \text{ km}, 0 \text{ km}, 0 \text{ km/s}, 3.5346866235 \text{ km/s}, 0 \text{ km/s}]}$

$\tau = tt^* = \boxed{9.3797565473 \times 10^6 \ s}$

@ Initial state, s/c from earth → $\vec{r}_1 = (x + \mu), 0, 0$

convert μ to dimensions → $\mu \cdot l^* = 4.6706846413 \times 10^3$ km

$\boxed{\vec{r}_1 = 5.0798684641 \times 10^4 \ \hat{x}, 0 \ \hat{y}, 0 \ \hat{z} \ \text{km}}$

For non-dimensional terms, period is 2π

$\therefore t = 25 \rightarrow \frac{25}{2\pi} = \boxed{3.9788735773 \text{ periods}}$

Problem 3
Part a

```
% Set options for ODE45
options = odeset('RelTol', 1e-12, 'AbsTol', 1e-12, 'Events', @(t,y) eventFn(t, y));

% Call ODE45 function
[tout, xout] = ode45(@(t, state)CR3BP(state, mu), [0 int_time], state0, options);

function state_dot = CR3BP(state, mu)
    % Circular Restricted 3 Body Problem non-dimensional EOMs
    x = state(1);
    y = state(2);
    z = state(3);
    xdot = state(4);
    ydot = state(5);
    zdot = state(6);

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

    state_dot(1, 1) = xdot;
    state_dot(2, 1) = ydot;
    state_dot(3, 1) = zdot;

    state_dot(4, 1) = 2*ydot + x - (1 - mu)*(x + mu)/(r1^3) - mu * (x - 1 + mu)/(r2^3);
    state_dot(5, 1) = -2*xdot + y - (1 - mu)*y/(r1^3) - mu*y/(r2^3);
    state_dot(6, 1) = - (1 - mu)*z/(r1^3) - mu*z/(r2^3);
end

function [value,isterminal,direction] = eventFn(t,y)
    value = y(2); % Want y to be 0
    isterminal = 1; % Halt integration when value is 0
    direction = 1; % When zero is approached from +ve i.e. y_dot > 0
end
```

Part b

State vector at the stopping condition:

$\bar{x} = [0.4175833509,\ 0,\ 0,\ 1.1166886815,\ 0.6908164723,\ 0]$ non-dimensional units

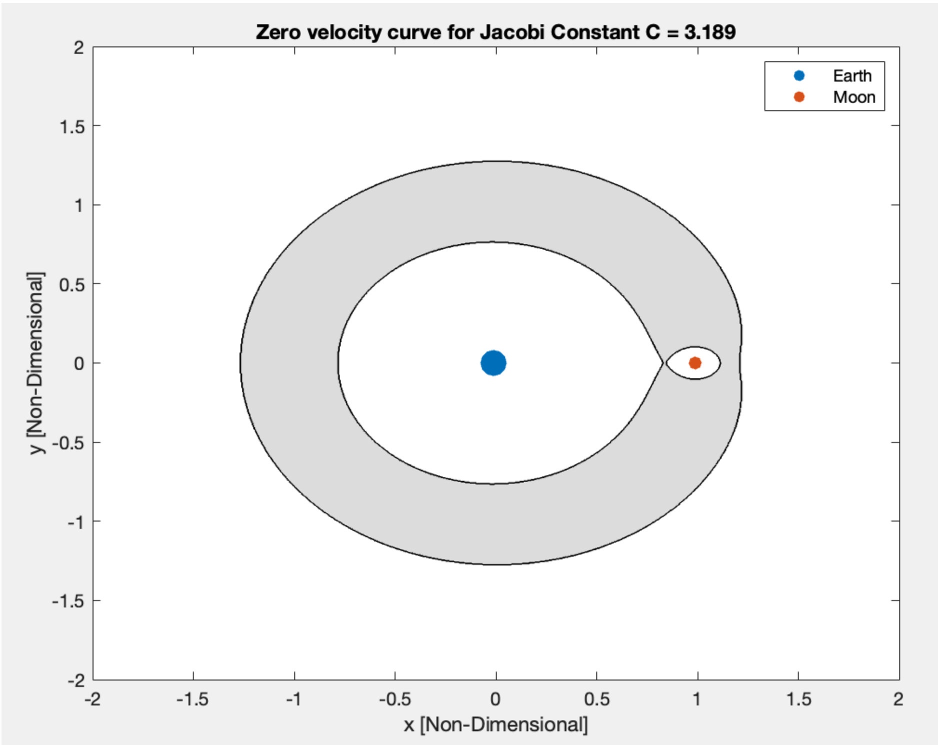
Integration time at stopping condition - 2.2690496918 non-dimensional units

Problem 4
Part a
Firstly, x and y are sampled between -2 and 2 which are arbitrarily picked. Then, a zvc matrix of the size of x by y is initialized with ones. Then, parsing through each point, 2U* is calculated and if 2U* is less than the given jacobi constant, that point is not allowable motion and set to -1. The reasoning for this logic is: C = 2U* - v^2 where v is 0 for zero-velocity curves. And, as seen in lecture, C(L1) > C(L2) and so on which means that when C is decreased, energy increases and allowable motion increases. That means that when 2U* is less than given C, it is trying to access motion at a higher energy level which is not accessible by the given jacobi constant. Hence, that space (or point) is non allowable motion.

```
mu = mass_ratio_em;
n = 5001;
x = linspace(-2, 2, n);
y = linspace(-2, 2, n);

c_given = 3.189;

zvc = full_zvc(c_given, mu, x, y);

function zvc = full_zvc(c_given, mu, x, y)
    % Initialize zero-velocity curve as a ones matrix
    zvc = ones([length(x), length(y)]);

    % Parse through x and y and calculate 2U_star at each point
    for i = 1:length(x)
        for j = 1:length(y)
            c_calc = u_star_times_2(x(i), y(j), mu);
            % If 2U_star is less than C, then that point is not allowable
            % motion. Assign zvc at that point to -1
            if c_calc < c_given
                zvc(i, j) = -1;
            end
        end
    end
end

function out = u_star_times_2(x, y, mu)
    r1 = sqrt((x + mu)^2 + y^2);
    r2 = sqrt((x - 1 + mu)^2 + y^2);
    out = (x^2 + y^2) + 2*(1 - mu)/r1 + 2*mu/r2;
end
```

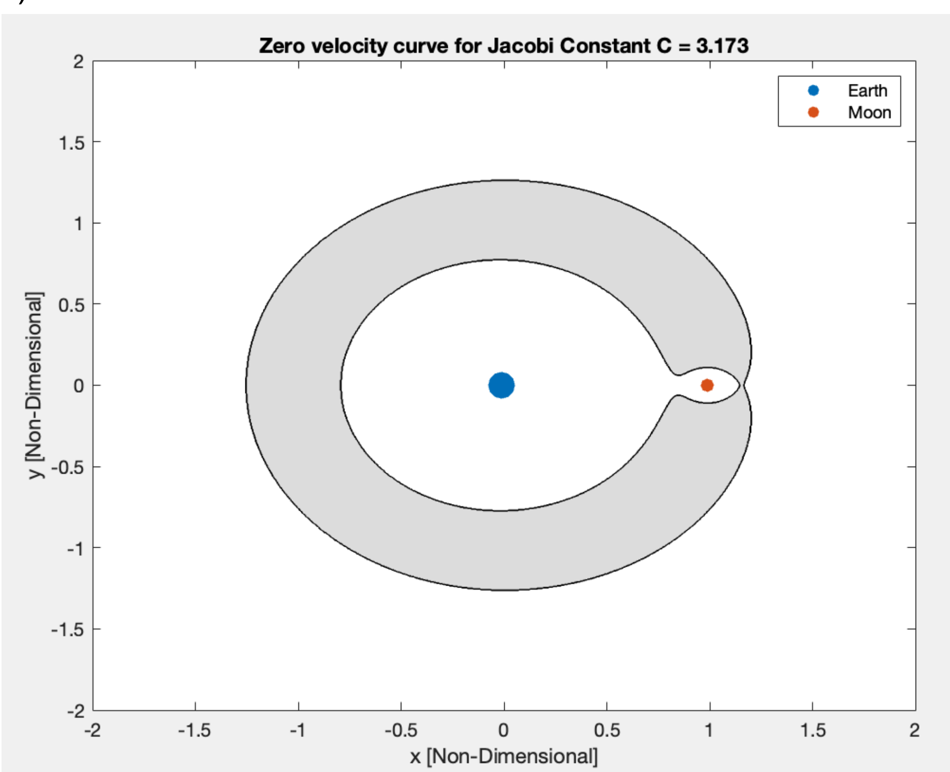
Part b

1)



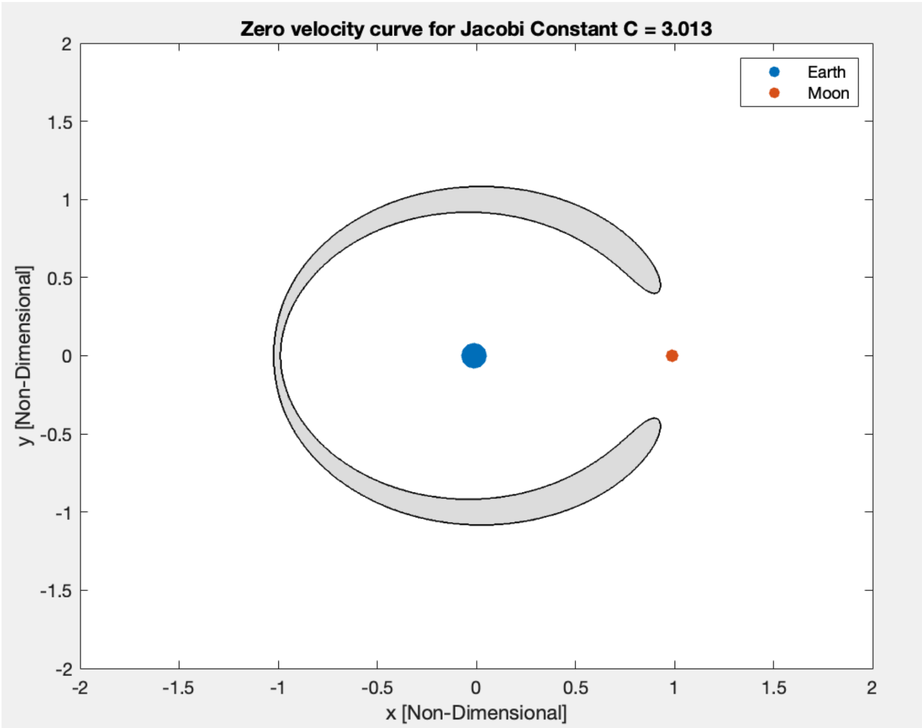
Zero velocity curve for Jacobi Constant C = 3.189
Earth
Moon
y [Non-Dimensional]
x [Non-Dimensional]
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2

2)



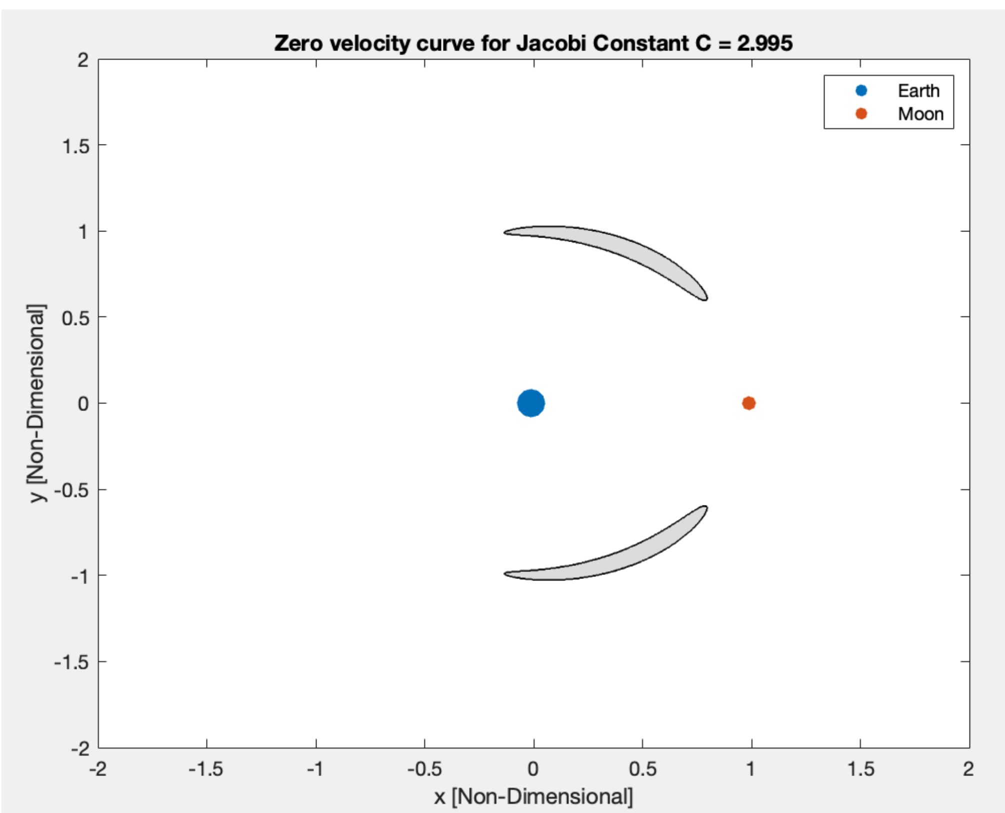
Zero velocity curve for Jacobi Constant C = 3.173
Earth
Moon
y [Non-Dimensional]
x [Non-Dimensional]
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2

3)



Zero velocity curve for Jacobi Constant C = 3.013
Earth
Moon
y [Non-Dimensional]
x [Non-Dimensional]
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2

4)



Zero velocity curve for Jacobi Constant C = 2.995
Earth
Moon
y [Non-Dimensional]
x [Non-Dimensional]
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2

Part c

All the discussions below are for natural trajectories and refer to ZVCs presented in lecture slides.

- C = 3.189
  - This energy level approximately matches ZVC at C(L1) because the L1 and L2 gateways are closed and if a trajectory existed around Earth or Moon, it would keep orbiting the respective body and not escape.
- C = 3.173
  - This energy level approximately matches ZVC at C(L2) because the L1 gateway is open and L2 gateway is closed and a trajectory could transfer between the earth moon system but not escape into the exterior region.
- C = 3.013
  - This energy level approximately matches ZVC at C(L3) because the L1 and L2 gateways are open and if a trajectory existed around Earth or Moon, it could escape into the exterior region through the L2 gateway.
- C = 2.995
  - This energy level approximately matches ZVC at C(L4/L5) because the L1 and L2 gateways are open and the “non-allowable motion” is around the L4 and L5 points. This Jacobi constant is most likely higher than C(L4/L5) because C(L4/L5) has no “non-allowable motion”.

Part d

In order to estimate equilibrium points, the jacobi constant is sampled from 2 to 3.2 (these bounds were picked after guessing and checking various values). Then, ZVCs are calculated for each jacobi constant. C(L4/L5) is picked when “non-allowable motion” first appears in the ZVCs. C(L3) is picked when there’s non-allowable motion to the left of P1 (earth) and y = 0. C(L2) is picked when there’s non-allowable motion to the right of P2 (moon) and y = 0. C(L1) is picked when there’s non-allowable motion between P1 and P2 and y = 0. Using this strategy, the following values of Jacobi constant are estimated:

Equilibrium Point	C	Position (x, y, z)
L1	3.19	(0.84, 0, 0)
L2	3.17	(1.16, 0, 0)
L3	3.01	(-1.05, 0, 0)
L4	2.99	(0.48, 0.87, 0)
L5	2.99	(0.48, -0.87, 0)

The position of lagrange points is then estimated visually from the ZVC contour plots.

```
clear; clc; close all;

% ASEN 6060 - HW 1, Problem 1
% Jash Bhalavat

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

% Sun
mu_sun = 132712440041.279419; % km3/s2
mass_sun = mu_sun / G; % kg
```

Part a

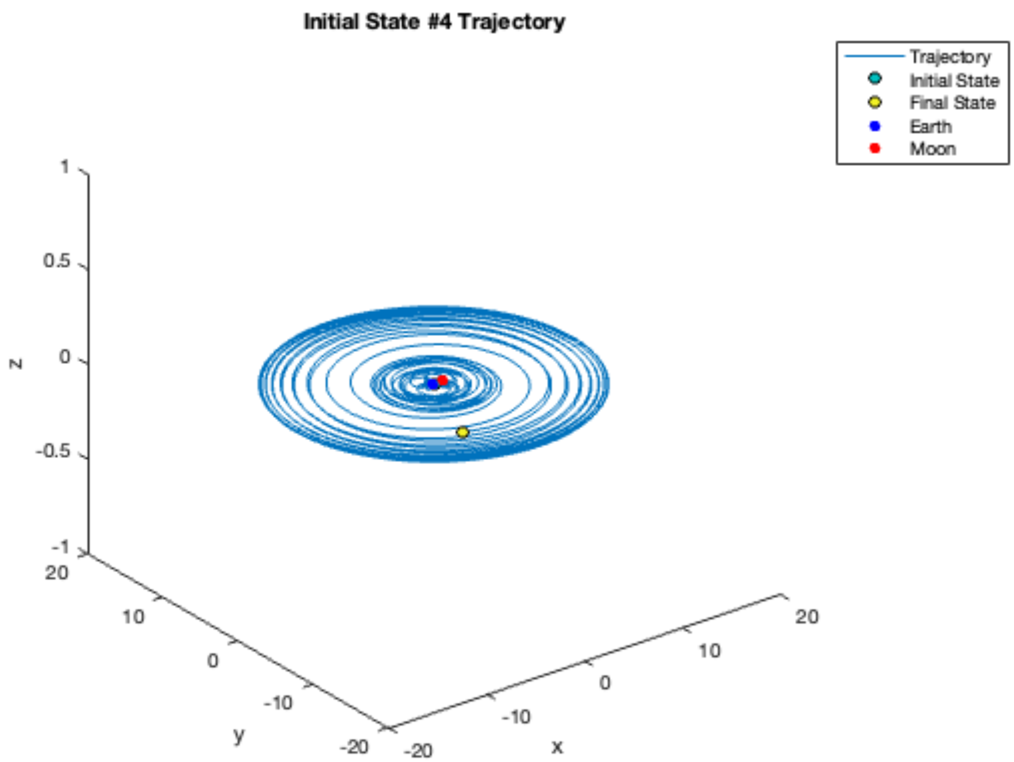
Earth-Moon system

```
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));

% Sun-Earth system
mass_ratio_se = mass_earth / (mass_earth + mass_sun);
m_star_se = mass_earth + mass_sun;
l_star_se = a_earth;
t_star_se = sqrt(l_star_se^3/(G * m_star_se));
```

Published with MATLAB® R2024a

```matlab
clear; clc; close all;

% ASEN 6060 - HW 1, Problem 2
% Jash Bhalavat
% 01/28/2025

% Constants
G = 6.67408 * 10^-11; % m3/(kgs2)
G = G / (10^9); % km3/(kgs2)
% Earth
mu_earth = 398600.435507; % km3/s2
mass_earth = mu_earth / G; % kg
% Moon
mu_moon = 4902.800118; % km3/s2
a_moon = 384400; % km
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;
mu = mass_ratio_em;

init_state_case = 4;

switch init_state_case
    case 1
        % Integration time [unitless]
        int_time = 2;
        % Initial State [unitless]
        state0 = [0.98, 0, 0, 0, 1.2, 0];
    case 2
        int_time = 8;
        state0 = [0.98, 0, 0, 0, 1.7, 0];
    case 3
        int_time = 25;
        state0 = [0.12, 0, 0, 0, 3.45, 0];
    case 4
        int_time = 250;
        state0 = [0.12, 0, 0, 0, 3.48, 0];
end

% Set options for ODE45
options = odeset('RelTol', 1e-12, 'AbsTol', 1e-12);

% Call ODE45 function
[tout, xout] = ode45(@(t, state)CR3BP(state, mu), [0 int_time], state0,
options);

function state_dot = CR3BP(state, mu)
    % Circular Restricted 3 Body Problem non-dimensional EOMs
    x = state(1);
    y = state(2);
    z = state(3);
```

```
    xdot = state(4);
    ydot = state(5);
    zdot = state(6);

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

    state_dot(1, 1) = xdot;
    state_dot(2, 1) = ydot;
    state_dot(3, 1) = zdot;

    state_dot(4, 1) = 2*ydot + x - (1 - mu)*(x + mu)/(r1^3) - mu * (x - 1 +
mu)/(r2^3);
    state_dot(5, 1) = -2*xdot + y - (1 - mu)*y/(r1^3) - mu*y/(r2^3);
    state_dot(6, 1) = - (1 - mu)*z/(r1^3) - mu*z/(r2^3);
end

% Plot
figure()
plot3(xout(:,1), xout(:,2), xout(:,3))
hold on
scatter3(xout(1,1), xout(1,2), xout(1,3), 'MarkerEdgeColor','k',
'MarkerFaceColor',[0 .75 .75])
scatter3(xout(end,1), xout(end,2), xout(end,3), 'MarkerEdgeColor','k',
'MarkerFaceColor',[0.95 .95 .15])
scatter3(-mu, 0, 0, 'filled', 'blue')
scatter3(1-mu, 0, 0, 'filled', 'red')
hold off
legend("Trajectory", "Initial State", "Final State", "Earth", "Moon")
title("Initial State #" + init_state_case + " Trajectory")
xlabel("x")
ylabel("y")
zlabel("z")
```



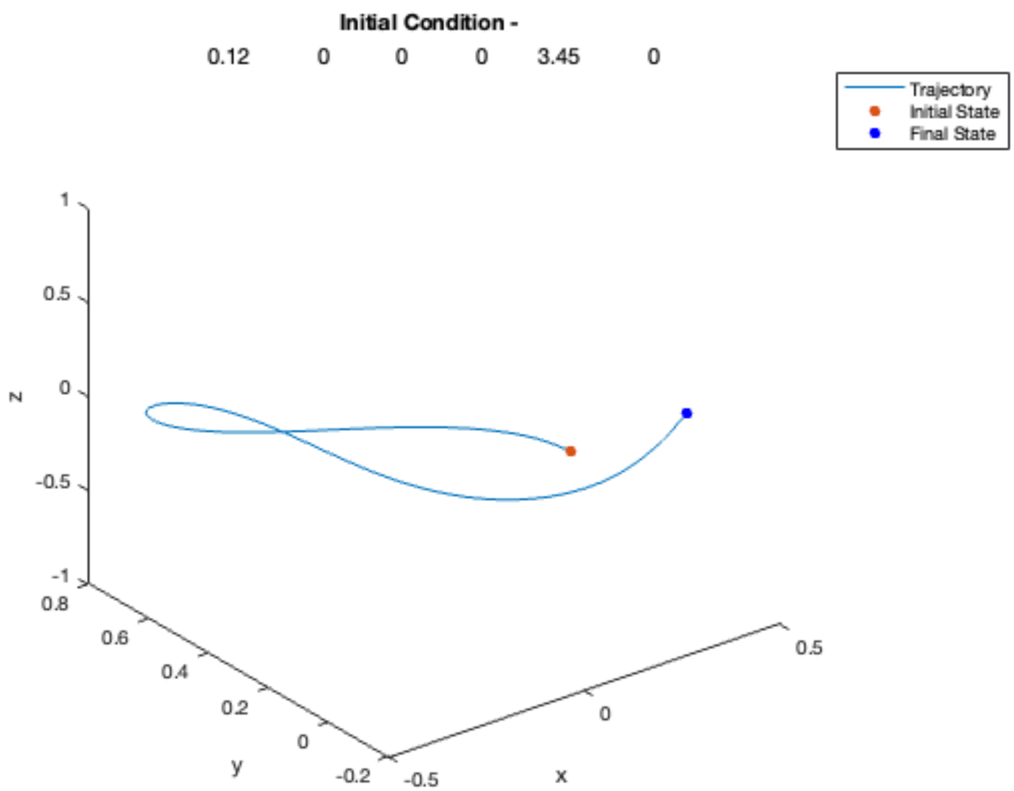
Part d

```
int_time = 25;
state0 = [0.12, 0, 0, 0, 3.45, 0];
l_star = a_moon;
t_star = (l_star^3/(G*m_star_em))^(1/2);
initial_pos_dim = state0(1:3) * l_star;
initial_vel_dim = state0(4:6) * l_star/t_star;
tau = int_time * t_star;

% Planet pos
pos_p1 = mu * l_star;
sc_at_t0_from_earth = initial_pos_dim + [pos_p1, 0, 0];

t_terms_of_period = int_time / (2*pi);
```

Published with MATLAB® R2024a

```
clear; clc; close all;

% ASEN 6060 - HW 1, Problem 3
% Jash Bhalavat
% 01/28/2025

% Constants
G = 6.67408 * 10^-11; % m3/(kgs2)
G = G / (10^9); % km3/(kgs2)
% Earth
mu_earth = 398600.435507; % km3/s2
mass_earth = mu_earth / G; % kg
% Moon
mu_moon = 4902.800118; % km3/s2
a_moon = 384400; % km
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;
mu = mass_ratio_em;

int_time = 25;
state0 = [0.12, 0, 0, 0, 3.45, 0];

% Set options for ODE45
options = odeset('RelTol', 1e-12, 'AbsTol', 1e-12, 'Events', @(t,y)
eventFn(t, y));

% Call ODE45 function
[tout, xout] = ode45(@(t, state)CR3BP(state, mu), [0 int_time], state0,
options);

function state_dot = CR3BP(state, mu)
    % Circular Restricted 3 Body Problem non-dimensional EOMs
    x = state(1);
    y = state(2);
    z = state(3);
    xdot = state(4);
    ydot = state(5);
    zdot = state(6);

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

    state_dot(1, 1) = xdot;
    state_dot(2, 1) = ydot;
    state_dot(3, 1) = zdot;

    state_dot(4, 1) = 2*ydot + x - (1 - mu)*(x + mu)/(r1^3) - mu * (x - 1 +
mu)/(r2^3);
    state_dot(5, 1) = -2*xdot + y - (1 - mu)*y/(r1^3) - mu*y/(r2^3);
    state_dot(6, 1) = - (1 - mu)*z/(r1^3) - mu*z/(r2^3);
```

```
end

function [value,isterminal,direction] = eventFn(t,y)
    value = y(2); % Want y to be 0
    isterminal = 1; % Halt integration when value is 0
    direction = 1; % When zero is approached from +ve i.e. y_dot > 0
end

% Plot
figure()
plot3(xout(:,1), xout(:,2), xout(:,3))
hold on
scatter3(xout(1,1), xout(1,2), xout(1,3), 'filled')
scatter3(xout(end,1), xout(end,2), xout(end,3), 'filled', 'blue')
hold off
legend("Trajectory", "Initial State", "Final State")
title("Initial Condition - ", num2str(state0) )
xlabel("x")
ylabel("y")
zlabel("z")
```



Published with MATLAB® R2024a

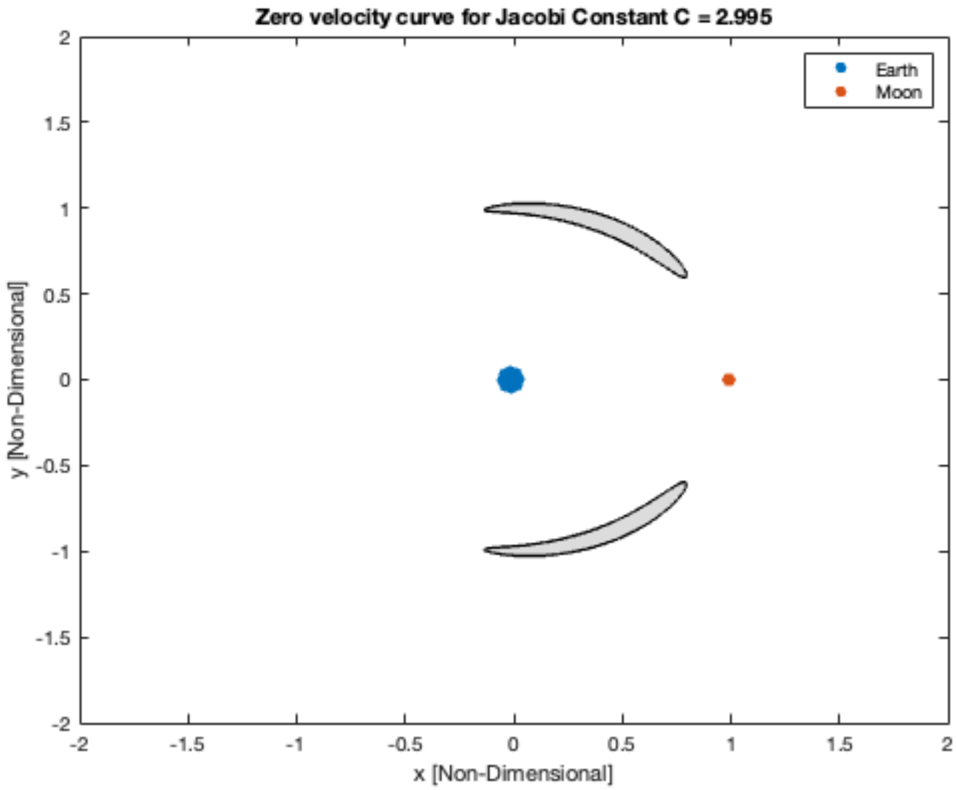
```
clear; clc; close all;

% ASEN 6060 - HW 1, Problem 4
% Jash Bhalavat
% 01/28/2025
```

Constants

```
G = 6.67408 * 10^-11; % m3/(kgs2)
G = G / (10^9); % km3/(kgs2)
% Earth
mu_earth = 398600.435507; % km3/s2
mass_earth = mu_earth / G; % kg
% Moon
mu_moon = 4902.800118; % km3/s2
a_moon = 384400; % km
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;
mu = mass_ratio_em;
```

Zero Velocity Surfaces

```
mu = mass_ratio_em;

n = 1001;
x = linspace(-2, 2, n);
y = linspace(-2, 2, n);

% Given Jacobi constant
c_case = 4;

switch c_case
    case 1
        c_given = 3.189;
    case 2
        c_given = 3.173;
    case 3
        c_given = 3.013;
    case 4
        c_given = 2.995;
end

zvc = full_zvc(c_given, mu, x, y);
plot_zvc(zvc, x, y, c_given, mu)
```



Part d

```
c_test = linspace(2,3.2,1000);

% Y index for zero
y_zero_ind = find(y == 0);

% Find -mu and 1 - mu X indices
[del, x_p1_ind] = min(abs(x - (-mu)));
[del, x_p2_ind] = min(abs(x - (1-mu)));

l1_calc = false;
l2_calc = false;
l3_calc = false;
l45_calc = false;

for i = 1:length(c_test)
    zvc = full_zvc(c_test(i), mu, x, y);

    % Check for C(L4/5)
    % Check if zvc has any dynamically excluded motion
    zvc_neg_1 = find(zvc == -1);
    if and(~isempty(zvc_neg_1), ~l45_calc)
        c_l45 = c_test(i-1);
        zvc_l45 = full_zvc(c_l45, mu, x, y);
```

```
        zvc_l45_minus_1 = full_zvc(c_test(i), mu, x, y);
        l45_calc = true;
    end

    % Check for C(L3)
    % L3 is on y = 0
    % Check where dynamically excluded motion first appears in that row
    zvc_zero_y = zvc(:, y_zero_ind);
    zvc_zero_y_neg_1 = find(zvc_zero_y == -1);
    if and(~isempty(zvc_zero_y_neg_1), ~l3_calc)
        c_l3 = c_test(i);
        zvc_l3 = full_zvc(c_l3, mu, x, y);
        l3_calc = true;
    end

    % Check for C(l2)
    % L2 is located beyond P2 on y=0
    zvc_beyond_p2 = zvc(x_p2_ind:end, y_zero_ind);
    zvc_beyond_p2_neg_1 = find(zvc_beyond_p2 == -1);
    if and(~isempty(zvc_beyond_p2_neg_1), ~l2_calc)
        c_l2 = c_test(i);
        zvc_l2 = full_zvc(c_l2, mu, x, y);
        l2_calc = true;
    end

    % Check for C(l1)
    % L1 is located between P1 and P2 on y=0
    zvc_between_p1_p2 = zvc(x_p1_ind:x_p2_ind, y_zero_ind);
    zvc_between_p1_p2_neg_1 = find(zvc_between_p1_p2 == -1);
    if and(~isempty(zvc_between_p1_p2_neg_1), ~l1_calc)
        c_l1 = c_test(i);
        zvc_l1 = full_zvc(c_l1, mu, x, y);
        l1_calc = true;
    end

end

plot_zvc(zvc_l1, x, y, c_l1, mu)
plot_zvc(zvc_l2, x, y, c_l2, mu)
plot_zvc(zvc_l3, x, y, c_l3, mu)
plot_zvc(zvc_l45, x, y, c_l45, mu)
plot_zvc(zvc_l45_minus_1, x, y, c_l45, mu)

function zvc = full_zvc(c_given, mu, x, y)
    zvc = ones([length(x), length(y)]);

    for i = 1:length(x)
        for j = 1:length(y)
            c_calc = u_star_times_2(x(i), y(j), mu);
            if c_calc < c_given
                zvc(i, j) = -1;
            end
        end
    end
```

```
end

function plot_zvc(zvc, x, y, c, mu)
    figure()
    contourf(x, y, zvc')
    map = [220/255, 220/255, 220/255
        1, 1, 1];
    colormap(map)
    hold on
    scatter(-mu, 0, 200, 'filled')
    scatter(1-mu, 0, 50, 'filled')
    hold off
    legend("", "Earth", "Moon")
    xlabel("x [Non-Dimensional]")
    ylabel("y [Non-Dimensional]")
    title("Zero velocity curve for Jacobi Constant C = " + num2str(c))
end

function out = u_star_times_2(x, y, mu)
    r1 = sqrt((x + mu)^2 + y^2);
    r2 = sqrt((x - 1 + mu)^2 + y^2);
    out = (x^2 + y^2) + 2*(1 - mu)/r1 + 2*mu/r2;
end
```

Warning: Contour not rendered for constant ZData

Published with MATLAB® R2024a