Table of Contents


```
clear; clc; close all;

% ASEN 6060 - HW 5, Prob 2a
% 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;

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

global count poincare_stored
```

Part a

```
TOL = 1e-12;
% Set options for ode113
options = odeset('RelTol', TOL, 'AbsTol', TOL);
```

```
% 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
l1_pos = [em_eq_pts(1,:), 0];
l2_pos = [em_eq_pts(2,:), 0];

x0_1 = [0.8213849,0,0,0,0.1475143,0];
x0_2 = [1.164855,0,0,0, -0.0516671,0];
T1 = 2.763299;
T2 = 3.377214;

V0_1 = [x0_1, T1]';
V0_2 = [x0_2, T2]';

L1_periodic = gen_3d_periodic_orbit_single_shooting(V0_1, mu, false);
[L1_tout, L1_xout] = ode113(@(t, state)CR3BP_full(state, mu), [0,
L1_periodic(end)], [L1_periodic(1:6); reshape(eye(6), [36,1])], options);

L2_periodic = gen_3d_periodic_orbit_single_shooting(V0_2, mu, false);
[L2_tout, L2_xout] = ode113(@(t, state)CR3BP_full(state, mu), [0,
L2_periodic(end)], [L2_periodic(1:6); reshape(eye(6), [36,1])], options);
```

Part b, c

```
n_crossings = 2;

part_b(L1_tout, L1_xout, mu, l1_pos, 10, n_crossings);
poincare_unstable = poincare_stored;

part_c(L2_tout, L2_xout, mu, l2_pos, 6, n_crossings);

title("Moon-Bound Stable/Unstable Manifolds associated with L1, L2 Lyapunov
Orbits")

figure(2)
scatter(poincare_unstable(:,2), poincare_unstable(:,1), 10, 'filled', 'red');
hold on
scatter(poincare_stored(:,2), poincare_stored(:,1), 10, 'filled', 'blue');
xlabel("$\dot{y}$", 'Interpreter','latex')
ylabel("y")
title("Poincar\'e Map", 'Interpreter','latex')
grid on
legend("Unstable", "Stable")
```

Functions

```
function part_b(tout, xout, mu, l1_pos, manifold_time, n_crossings)
    % Set options for ode113()
    % Part b
    options = odeset('RelTol', 1e-12, 'AbsTol', 1e-12, 'Events', @(t,y)
eventFn(t, y, mu));
```

```
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])';

moon_unstable_cnt = 0;

% Begin for loop
for i = 1: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(abs(evals_minus_1), 2);

    % If eval is real and not trivial, assign stable and unstable
    % indices
    for j = 1:2
        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 unstable evec and normalize eigenvector by 1st 3 terms
```

```
        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;

        % ONLY FOR L1
        % If x-velocity is positive, moon-bound
        % If x-velocity if negative, earth-bound
        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_unstable_t, moon_unstable_x] = ode113(@(t, state)CR3BP(state,
mu), [0, manifold_time], moon_unstable, options);
        [earth_unstable_t, earth_unstable_x] = ode113(@(t, state)CR3BP(state,
mu), [0, manifold_time], earth_unstable, options);

        % plot(moon_unstable_x(:,1), moon_unstable_x(:,2), 'red')
        % plot(earth_unstable_x(:,1), earth_unstable_x(:,2), 'red')

        if abs(moon_unstable_x(end,1) - (1-mu)) < 1e-6
            moon_unstable_cnt = moon_unstable_cnt + 1;
            moon_bound_unstable(:,moon_unstable_cnt) = moon_unstable;
        elseif abs(earth_unstable_x(end,1) - (1-mu)) < 1e-6
            moon_unstable_cnt = moon_unstable_cnt + 1;
            moon_bound_unstable(:,moon_unstable_cnt) = earth_unstable;
        end

    end

    global count;
    global poincare_stored;
    poincare_stored = [];
    for k = 1:moon_unstable_cnt
        count = 0;
        options = odeset('RelTol', 1e-12, 'AbsTol', 1e-12, 'Events', @(t,y)
b_eventFn(t, y, mu, n_crossings));
        [moon_unstable_t, moon_unstable_x] = ode113(@(t, state)CR3BP(state,
mu), [0, manifold_time], moon_bound_unstable(:,k), options);
        plot(moon_unstable_x(:,1), moon_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 part_c(tout, xout, mu, l2_pos, manifold_time, n_crossings)
    % Set options for ode113()
    % Part c
    options = odeset('RelTol', 1e-12, 'AbsTol', 1e-12, 'Events', @(t,y)
eventFn(t, y, mu));

    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(l2_pos(1), l2_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])';

    moon_stable_cnt = 0;

    % Begin for loop
    for i = 1: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(abs(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);

        % 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;

        % 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

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

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

        if (abs(moon_stable_x(end,1) - (1-mu)) < 1e-6 && moon_stable_x(end,2)
< 0)
            moon_stable_cnt = moon_stable_cnt + 1;
            moon_bound_stable(:,moon_stable_cnt) = moon_stable;
        else
            moon_stable_cnt = moon_stable_cnt + 1;
            moon_bound_stable(:,moon_stable_cnt) = earth_stable;
        end
    end

    global count;
    global poincare_stored;
    poincare_stored = [];
    for k = 1:moon_stable_cnt
        count = 0;
        options = odeset('RelTol', 1e-12, 'AbsTol', 1e-12, 'Events', @(t,y)
```

```
c_eventFn(t, y, mu, n_crossings));
        [moon_stable_t, moon_stable_x] = ode113(@(t, state)CR3BP(state, mu),
[0, -manifold_time], moon_bound_stable(:,k), options);
        plot(moon_stable_x(:,1), moon_stable_x(:,2), 'blue')
    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] = c_eventFn(t,y,mu,n_crossings)
    global count;
    global poincare_stored;
    if count < n_crossings
        value = y(1) - (1-mu);
        isterminal = 0;
        direction = -1;
        if (abs(value) < 1e-12 && y(4) > 0)
            count = count + 1;
            poincare_stored = [poincare_stored; y(2), y(5)];

        end
    elseif count == n_crossings
        value = y(1) - (1-mu); % Want x to be 1-mu
        isterminal = 1; % Halt integration when value is 0
        direction = -1; % When zero is approached from +ve i.e. x_dot > 0
        if (abs(value) < 1e-12 && y(4) > 0)
            poincare_stored = [poincare_stored; y(2), y(5)];

        end
    end
end


function [value,isterminal,direction] = b_eventFn(t,y,mu,n_crossings)
    global count;
    global poincare_stored;
    if count < n_crossings
        value = y(1) - (1-mu);
        isterminal = 0;
        direction = 1;
        if (abs(value) < 1e-12 && y(4) > 0)
            count = count + 1;
            poincare_stored = [poincare_stored; y(2), y(5)];

        end
    elseif count == n_crossings
        value = y(1) - (1-mu); % Want x to be 1-mu
        isterminal = 1; % Halt integration when value is 0
        direction = 1; % When zero is approached from +ve i.e. x_dot > 0
        if (abs(value) < 1e-12 && y(4) > 0)
```

```
            poincare_stored = [poincare_stored; y(2), y(5)];

        end
    end
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
```

Published with MATLAB® R2024a