Table of Contents

```
clear; clc; close all;

% ASEN 6060 - Final Project
% 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];

% 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];

TOL = 5e-14;

% Set options for ode113
options = odeset('RelTol', TOL, 'AbsTol', TOL);

global count poincare_stored iteration
```

1A & 1B Orbits

```
% L1 orbit is a Lyapunov orbit
load("V_family_L1_Lyapunov.mat")
V_family_L1_Lyapunov = V_family;
l1_orbit_idx = 50;

% L2 orbit is a Lyapunov orbit
load("V_family_L2_Lyapunov.mat")
V_family_L2_Lyapunov = V_family_a1;
l2_orbit_idx = 10;

figure(1)
scatter(l1_pos(1), l1_pos(2), 'filled', 'blue')
hold on
scatter(l2_pos(1), l2_pos(2), 'filled', 'red')
scatter(p2_pos(1), p2_pos(2), 'filled', 'black')

[tout_L1, xout_L1] = ode113(@(t,state)CR3BP_full(state,
mu), [0, V_family_L1_Lyapunov(7,l1_orbit_idx)],
[V_family_L1_Lyapunov(1:6,l1_orbit_idx); reshape(eye(6), [36,1])], options);
plot(xout_L1(:,1), xout_L1(:,2), 'LineWidth',2)

[tout_L2, xout_L2] = ode113(@(t,state)CR3BP_full(state,
mu), [0, V_family_L2_Lyapunov(7,l2_orbit_idx)],
[V_family_L2_Lyapunov(1:6,l2_orbit_idx); reshape(eye(6), [36,1])], options);
plot(xout_L2(:,1), xout_L2(:,2), 'LineWidth',2)
```

```
cj_L1 = jacobi_constant(V_family_L1_Lyapunov(:,l1_orbit_idx), mu);
cj_L2 = jacobi_constant(V_family_L2_Lyapunov(:,l2_orbit_idx), mu);
delta_cj = cj_L2 - cj_L1;

hold off
legend("L1", "L2", "Moon", "L1 Lyapunov Orbit", "L2 Lyapunov Orbit")
xlabel('$$\hat{x}$$','Interpreter','Latex', 'FontSize',18)
ylabel('$$\hat{y}$$','Interpreter','Latex', 'FontSize',18)
grid on
axis equal
title("L1 and L2 Lyapunov Orbits (\DeltaC_j = " + abs(delta_cj) + ")")
```

Manifolds

```
n_crossings = 2;
L1_manifold_time = 10;
L2_manifold_time = 6;

L1_manifolds = unstable_manifolds(tout_L1, xout_L1, mu, l1_pos,
L1_manifold_time, n_crossings);
L2_manifolds = stable_manifolds(tout_L2, xout_L2, mu, l2_pos,
L2_manifold_time, n_crossings);

poincare_stored = [];
stable_options = odeset('RelTol', 1e-12, 'AbsTol', 1e-12, 'Events', @(t,y)
stable_event(t, y, mu, n_crossings));
options = odeset('RelTol', 1e-12, 'AbsTol', 1e-12);
options_event = odeset('RelTol', 1e-12, 'AbsTol', 1e-12, 'Events', @(t,y)
eventFn(t, y, mu));
unstable_options = odeset('RelTol', 1e-12, 'AbsTol', 1e-12, 'Events', @(t,y)
unstable_event(t, y, mu, n_crossings));

figure(2)
plot(xout_L1(:,1), xout_L1(:,2), 'black', 'LineWidth', 3)
hold on
plot(xout_L2(:,1), xout_L2(:,2), 'black', 'LineWidth', 3)
scatter(l2_pos(1), l2_pos(2), 'filled', 'red')
scatter(p2_pos(1), p2_pos(2), 'filled', ' black')

iteration = 1;

for k = 1:length(L1_manifolds)
    count = 0;
    [moon_unstable_t, moon_unstable_x] = ode113(@(t, state)CR3BP(state, mu),
[0, L1_manifold_time], L1_manifolds(:,k), unstable_options);
    unstable_end_times(k) = moon_unstable_t(end);
    unstable_end_states(k, :) = moon_unstable_x(end, :);
    plot(moon_unstable_x(:,1), moon_unstable_x(:,2), 'red')
    iteration = iteration + 1;
end
grid on
axis equal
xlabel('$$\hat{x}$$','Interpreter','Latex', 'FontSize',18)
```

```
ylabel('$$\hat{y}$$','Interpreter','Latex', 'FontSize',18)

poincare_unstable = poincare_stored;

poincare_stored = [];

iteration = 1;

for k = 1:length(L2_manifolds)
    count = 0;
    [moon_stable_t, moon_stable_x] = ode113(@(t, state)CR3BP(state, mu), [0,
-L2_manifold_time], L2_manifolds(:,k), stable_options);
    stable_end_times(k) = moon_stable_t(end);
    stable_end_states(k, :) = moon_stable_x(end, :);
    if (abs(stable_end_states(k, 1) - (1-mu)) < 1e-6 && stable_end_states(k,
2) < 0)
        plot(moon_stable_x(:,1), moon_stable_x(:,2), 'blue')
    end
    iteration = iteration + 1;
end
poincare_stable = poincare_stored;

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

Poincare Map

```
figure(3)
subplot(1,2,1)
scatter(poincare_unstable(:,2), poincare_unstable(:,1), 10, 'filled', 'red');
hold on
scatter(poincare_stable(:,2), poincare_stable(:,1), 10, 'filled', 'blue');
xlabel("$\dot{y}$", 'Interpreter','latex')
ylabel("y")
title("Poincar\'e Map", 'Interpreter','latex')
grid on

stable_y_value = -0.043753;
stable_ydot_value = 0.00922752;
[pc_stable_min, pc_stable_min_ind] = min(abs(poincare_stable(:,1) -
stable_y_value));
stable_ig = poincare_stable(pc_stable_min_ind, 3);


unstable_y_value = -0.0437569;
unstable_ydot_value = 0.00797096;
[pc_unstable_min, pc_unstable_min_ind] = min(abs(poincare_unstable(:,1) -
unstable_y_value));
unstable_ig = poincare_unstable(pc_unstable_min_ind, 3);

stable_y_value_2 = -0.0441014;
stable_ydot_value_2 = -0.155138;
```

```
[pc_stable_min_2, pc_stable_min_ind_2] = min(abs(poincare_stable(:,1) -
stable_y_value_2));
stable_ig_2 = poincare_stable(pc_stable_min_ind_2, 3);

scatter(poincare_stable(pc_stable_min_ind_2, 2),
poincare_stable(pc_stable_min_ind_2, 1), 'filled', 'green')
scatter(poincare_stable(pc_stable_min_ind, 2),
poincare_stable(pc_stable_min_ind, 1), 'filled', 'cyan')
scatter(poincare_unstable(pc_unstable_min_ind, 2),
poincare_unstable(pc_unstable_min_ind, 1), 'filled', 'black')
legend("Unstable", "Stable", "Initial Guess - Stable (1A)", "Initial Guess -
Stable (1B)", "Initial Guess - Unstable")
hold off

subplot(1,2,2)
scatter(poincare_unstable(:,2), poincare_unstable(:,1), 10, 'filled', 'red');
hold on
scatter(poincare_stable(:,2), poincare_stable(:,1), 10, 'filled', 'blue');
xlabel("$\dot{y}$", 'Interpreter','latex')
ylabel("y")
title("Poincar\'e Map (Zoomed in)", 'Interpreter','latex')
grid on
scatter(poincare_stable(pc_stable_min_ind_2, 2),
poincare_stable(pc_stable_min_ind_2, 1), 'filled', 'green')
scatter(poincare_stable(pc_stable_min_ind, 2),
poincare_stable(pc_stable_min_ind, 1), 'filled', 'cyan')
scatter(poincare_unstable(pc_unstable_min_ind, 2),
poincare_unstable(pc_unstable_min_ind, 1), 'filled', 'black')
legend("Unstable", "Stable", "Initial Guess - Stable (1A)", "Initial Guess -
Stable (1B)", "Initial Guess - Unstable")
hold off
xlim([-0.25, 0.05])
ylim([-0.05, -0.04])

sgtitle("Transfers 1A & 1B")
```

Multiple Shooting Initial Guess - 1

```
% Initial guess
x_1_0 = xout_L1(1, 1:6)';
delta_T1 = tout_L1(unstable_ig);
[tout, x_1] = ode113(@(t, state)CR3BP(state, mu), [0, delta_T1], x_1_0);
x_1_f = xout_L1(unstable_ig, 1:6)';

x_2_0 = L1_manifolds(:, unstable_ig);
delta_T2 = unstable_end_times(unstable_ig);
[tout, x_2] = ode113(@(t, state)CR3BP(state, mu), [0, delta_T2], x_2_0);
x_2_f = unstable_end_states(unstable_ig, :)';

x_3_f = L2_manifolds(:, stable_ig);
delta_T3 = -stable_end_times(stable_ig);
x_3_0 = stable_end_states(stable_ig, :)';
```

```
x_4_0 = xout_L2(stable_ig, 1:6)';
delta_T4 = V_family_L2_Lyapunov(7, l2_orbit_idx) - tout_L2(stable_ig);
[tout, x_4] = ode113(@(t, state)CR3BP(state, mu), [0, delta_T4], x_4_0);
x_4_f = x_4(end, 1:6)';

r_des_4 = xout_L2(1, 1:3)';

V0 = [x_1_0; delta_T1; x_2_0; delta_T2; x_3_0; delta_T3; x_4_0; delta_T4];

e1 = x_1_f - x_1_0;
e2 = x_2_f - x_2_0;
e3 = x_3_f - x_3_0;

V_soln = multiple_shooting(V0, mu, true, r_des_4);

flight_time_1B = V_soln(7) + V_soln(14) + V_soln(21) + V_soln(28);
```

Delta V Calc

```
v_1_f = x_1_f(4:6);
v_2_0 = x_2_0(4:6);
v_2_f = x_2_f(4:6);
v_3_0 = x_3_0(4:6);
v_3_f = x_3_f(4:6);
v_4_0 = x_4_0(4:6);

function out = calc_dv(v1, v2)
    out = sqrt((v1(1) - v2(1))^2 + (v1(2) - v2(2))^2 + (v1(3) - v2(3))^2);
end

dv1 = calc_dv(v_1_f, v_2_0);
dv2 = calc_dv(v_2_f, v_3_0);
dv3 = calc_dv(v_3_f, v_4_0);

dv_tot_1B = dv1 + dv2 + dv3;
dv_tot_dim_1B = dv_tot_1B * l_star_em / t_star_em;

% x_1_0 = xout_L1(unstable_ig, 1:6)';
% ig_um = L1_manifolds(:, unstable_ig);
% ig_sm = L2_manifolds(:, stable_ig);
% ig_L2_PO = xout_L2(stable_ig, 1:6)';
```

Uncorrected Trajectory

```
[~, x_1] = ode113(@(t, state)CR3BP(state, mu), [0, V0(7)], V0(1:6), options);
[~, x_2] = ode113(@(t, state)CR3BP(state, mu), [0, V0(14)], V0(8:13),
options);
[~, x_3] = ode113(@(t, state)CR3BP(state, mu), [0, V0(21)], V0(15:20),
options);
[~, x_4] = ode113(@(t, state)CR3BP(state, mu), [0, V0(28)], V0(22:27),
options);

figure(5)
```

```
subplot(2, 1, 2)
scatter(l1_pos(1), l1_pos(2), 'filled', 'blue')
hold on
scatter(l2_pos(1), l2_pos(2), 'filled', 'red')
scatter(p2_pos(1), p2_pos(2), 'filled', 'black')

plot(x_1(:,1), x_1(:,2), 'LineWidth', 2)
plot(x_2(:,1), x_2(:,2), 'LineWidth', 2)
plot(x_3(:,1), x_3(:,2), 'LineWidth', 2)
plot(x_4(:,1), x_4(:,2), 'LineWidth', 2)
plot(xout_L1(:,1), xout_L1(:,2), 'black', 'LineWidth', 2)
plot(xout_L2(:,1), xout_L2(:,2), 'black',  'LineWidth', 2)

legend("L1", "L2", "Moon")
xlabel('$$\hat{x}$$','Interpreter','Latex', 'FontSize',18)
ylabel('$$\hat{y}$$','Interpreter','Latex', 'FontSize',18)
grid on
axis equal
title("1B")

sgtitle("1A and 1B Initial Guesses (Uncorrected) Transfers")
```

Corrected Trajectory

```
[~, x_1] = ode113(@(t, state)CR3BP(state, mu), [0, V_soln(7)], V_soln(1:6),
options);
[~, x_2] = ode113(@(t, state)CR3BP(state, mu), [0, V_soln(14)], V_soln(8:13),
options);
[~, x_3] = ode113(@(t, state)CR3BP(state, mu), [0, V_soln(21)],
V_soln(15:20), options);
[~, x_4] = ode113(@(t, state)CR3BP(state, mu), [0, V_soln(28)],
V_soln(22:27), options);

x1f = x_1(end,:);

figure(6)
subplot(2,1,2)
scatter(l1_pos(1), l1_pos(2), 'filled', 'blue')
hold on
scatter(l2_pos(1), l2_pos(2), 'filled', 'red')
scatter(p2_pos(1), p2_pos(2), 'filled', 'black')

plot(x_1(:,1), x_1(:,2), 'LineWidth', 2)
plot(x_2(:,1), x_2(:,2), 'LineWidth', 2)
plot(x_3(:,1), x_3(:,2), 'LineWidth', 2)
plot(x_4(:,1), x_4(:,2), 'LineWidth', 2)
plot(xout_L1(:,1), xout_L1(:,2), 'black', 'LineWidth', 2)
plot(xout_L2(:,1), xout_L2(:,2), 'black',  'LineWidth', 2)

hold off
legend("L1", "L2", "Moon")
xlabel('$$\hat{x}$$','Interpreter','Latex', 'FontSize',18)
ylabel('$$\hat{y}$$','Interpreter','Latex', 'FontSize',18)
```

```
grid on
axis equal
title("1B")

sgtitle("1A and 1B Corrected Transfers")
```

Multiple Shooting Initial Guess - 2

```
% Initial guess
x_1_0 = xout_L1(1, 1:6)';
delta_T1 = tout_L1(unstable_ig);
[tout, x_1] = ode113(@(t, state)CR3BP(state, mu), [0, delta_T1], x_1_0);
x_1_f = xout_L1(unstable_ig, 1:6)';

x_2_0 = L1_manifolds(:, unstable_ig);
delta_T2 = unstable_end_times(unstable_ig);
[tout, x_2] = ode113(@(t, state)CR3BP(state, mu), [0, delta_T2], x_2_0);
x_2_f = unstable_end_states(unstable_ig, :)';

x_3_f = L2_manifolds(:, stable_ig_2);
[tout, x_3] = ode113(@(t, state)CR3BP(state, mu), [0, -L2_manifold_time],
x_3_f, options_event);
% delta_T3 = -stable_end_times(stable_ig_2);
delta_T3 = -tout(end);
x_3_0 = x_3(end,:)';
% x_3_0 = stable_end_states(stable_ig_2, :)';


x_4_0 = xout_L2(stable_ig_2, 1:6)';
delta_T4 = V_family_L2_Lyapunov(7, l2_orbit_idx) - tout_L2(stable_ig_2);
[tout, x_4] = ode113(@(t, state)CR3BP(state, mu), [0, delta_T4], x_4_0);
x_4_f = x_4(end, 1:6)';

r_des_4 = xout_L2(1, 1:3)';

V0 = [x_1_0; delta_T1; x_2_0; delta_T2; x_3_0; delta_T3; x_4_0; delta_T4];

e1 = x_1_f - x_1_0;
e2 = x_2_f - x_2_0;
e3 = x_3_f - x_3_0;

V_soln = multiple_shooting(V0, mu, true, r_des_4);

flight_time_1A = V_soln(7) + V_soln(14) + V_soln(21) + V_soln(28);
```

Delta V Calc

```
v_1_f = x_1_f(4:6);
v_2_0 = x_2_0(4:6);
v_2_f = x_2_f(4:6);
v_3_0 = x_3_0(4:6);
v_3_f = x_3_f(4:6);
v_4_0 = x_4_0(4:6);
```

```
dv1 = calc_dv(v_1_f, v_2_0);
dv2 = calc_dv(v_2_f, v_3_0);
dv3 = calc_dv(v_3_f, v_4_0);

dv_tot_1A = dv1 + dv2 + dv3;
dv_tot_dim_1A = dv_tot_1A * l_star_em / t_star_em;


% x_1_0 = xout_L1(unstable_ig, 1:6)';
% ig_um = L1_manifolds(:, unstable_ig);
% ig_sm = L2_manifolds(:, stable_ig);
% ig_L2_PO = xout_L2(stable_ig, 1:6)';
```

Uncorrected Trajectory

```
[~, x_1] = ode113(@(t, state)CR3BP(state, mu), [0, V0(7)], V0(1:6), options);
[~, x_2] = ode113(@(t, state)CR3BP(state, mu), [0, V0(14)], V0(8:13),
options);
[~, x_3] = ode113(@(t, state)CR3BP(state, mu), [0, V0(21)], V0(15:20),
options);
[~, x_4] = ode113(@(t, state)CR3BP(state, mu), [0, V0(28)], V0(22:27),
options);

figure(5)
subplot(2, 1, 1)
scatter(l1_pos(1), l1_pos(2), 'filled', 'blue')
hold on
scatter(l2_pos(1), l2_pos(2), 'filled', 'red')
scatter(p2_pos(1), p2_pos(2), 'filled', 'black')

plot(x_1(:,1), x_1(:,2), 'LineWidth', 2)
plot(x_2(:,1), x_2(:,2), 'LineWidth', 2)
plot(x_3(:,1), x_3(:,2), 'LineWidth', 2)
plot(x_4(:,1), x_4(:,2), 'LineWidth', 2)

hold off
legend("L1", "L2", "Moon")
xlabel('$$\hat{x}$$','Interpreter','Latex', 'FontSize',18)
ylabel('$$\hat{y}$$','Interpreter','Latex', 'FontSize',18)
grid on
axis equal
title("1A")
```

Corrected Trajectory

```
[~, x_1] = ode113(@(t, state)CR3BP(state, mu), [0, V_soln(7)], V_soln(1:6),
options);
[~, x_2] = ode113(@(t, state)CR3BP(state, mu), [0, V_soln(14)], V_soln(8:13),
options);
[~, x_3] = ode113(@(t, state)CR3BP(state, mu), [0, V_soln(21)],
V_soln(15:20), options);
[~, x_4] = ode113(@(t, state)CR3BP(state, mu), [0, V_soln(28)],
V_soln(22:27), options);

x1f = x_1(end,:);
figure(6)
subplot(2,1,1)
scatter(l1_pos(1), l1_pos(2), 'filled', 'blue')
hold on
scatter(l2_pos(1), l2_pos(2), 'filled', 'red')
scatter(p2_pos(1), p2_pos(2), 'filled', 'black')

plot(x_1(:,1), x_1(:,2), 'LineWidth', 2)
plot(x_2(:,1), x_2(:,2), 'LineWidth', 2)
plot(x_3(:,1), x_3(:,2), 'LineWidth', 2)
plot(x_4(:,1), x_4(:,2), 'LineWidth', 2)
plot(xout_L1(:,1), xout_L1(:,2), 'black', 'LineWidth', 2)
plot(xout_L2(:,1), xout_L2(:,2), 'black',  'LineWidth', 2)

hold off
legend("L1", "L2", "Moon")
xlabel('$$\hat{x}$$','Interpreter','Latex', 'FontSize',18)
ylabel('$$\hat{y}$$','Interpreter','Latex', 'FontSize',18)
grid on
axis equal
title("1A")
```

Functions

```
function moon_bound_stable = stable_manifolds(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_stable_cnt = 0;
```

```
% Begin for loop
for i = 1:length(tout)
    print_out = sprintf('Discrete Time Step  - %d', i);
    disp(print_out)

    % 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
    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;

    % ONLY FOR L1
    % If x-velocity is positive, moon-bound
    % If x-velocity if negative, earth-bound
    x_manifold_s_p = xout(i,1:6)' + d * stable_evec;
    x_manifold_s_n = xout(i,1:6)' - d * stable_evec;
    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_unstable_x(:,1), moon_unstable_x(:,2), 'red')
        % plot(earth_unstable_x(:,1), earth_unstable_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;
        elseif abs(earth_stable_x(end,1) - (1-mu)) < 1e-6
            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_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 moon_bound_stable = unstable_manifolds(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_unstable_cnt = 0;

% Begin for loop
for i = 1:length(tout)
    print_out = sprintf('Discrete Time Step  - %d', i);
    disp(print_out)

    % 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
    [~, unstable_index] = max(real(evals_minus_1));

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

```
        % Step into manifold
        x_manifold_u_p = xout(i,1:6)' + d * unstable_evec;
        x_manifold_u_n = xout(i,1:6)' - d * unstable_evec;

        % If x-velocity is positive, moon-bound
        % If x-velocity if negative, earth-bound
        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_stable_x(:,1), moon_stable_x(:,2), 'blue')
        % plot(earth_stable_x(:,1), earth_stable_x(:,2), 'red')

        if (abs(moon_unstable_x(end,1) - (1-mu)) < 1e-6 &&
moon_unstable_x(end,2) < 0)
            moon_unstable_cnt = moon_unstable_cnt + 1;
            moon_bound_stable(:,moon_unstable_cnt) = moon_unstable;
        else
            moon_unstable_cnt = moon_unstable_cnt + 1;
            moon_bound_stable(:,moon_unstable_cnt) = earth_unstable;
        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] = unstable_event(t,y,mu,n_crossings)
    global count;
    global poincare_stored;
    global iteration;
    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), iteration];
            % counted_once = true;
        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), iteration];
            % counted_twice = true;
        end
    end
end

function [value,isterminal,direction] = stable_event(t,y,mu,n_crossings)
    global count;
    global poincare_stored;
    global iteration;
    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), iteration];
            % counted_once = true;
        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), iteration];
            % counted_twice = true;
        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

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

function C = jacobi_constant(x, mu)
    C = u_star_times_2(x(1), x(2), x(3), mu) - x(4)^2 - x(5)^2 - x(6)^2;
end

clear; clc; close all;

% ASEN 6060 - Final Project, 2A and 2B
% 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];

% 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];

TOL = 5e-14;

% Set options for ode113
options = odeset('RelTol', TOL, 'AbsTol', TOL);

global count poincare_stored iteration
```

2A & 2B Orbits

```
% L1 orbit is a Lyapunov orbit
load("V_family_L1_Lyapunov.mat")
V_family_L1_Lyapunov = V_family;
l1_orbit_idx = 50;

% L2 orbit is a Lyapunov orbit
load("V_family_L2_Lyapunov.mat")
V_family_L2_Lyapunov = V_family_a1;
l2_orbit_idx = 10;

figure(1)
scatter(l1_pos(1), l1_pos(2), 'filled', 'blue')
hold on
scatter(l2_pos(1), l2_pos(2), 'filled', 'red')
scatter(p2_pos(1), p2_pos(2), 'filled', 'black')

[tout_L1, xout_L1] = ode113(@(t,state)CR3BP_full(state,
mu), [0, V_family_L1_Lyapunov(7,l1_orbit_idx)],
[V_family_L1_Lyapunov(1:6,l1_orbit_idx); reshape(eye(6), [36,1])], options);
plot(xout_L1(:,1), xout_L1(:,2), 'LineWidth',2)

[tout_L2, xout_L2] = ode113(@(t,state)CR3BP_full(state,
mu), [0, V_family_L2_Lyapunov(7,l2_orbit_idx)],
[V_family_L2_Lyapunov(1:6,l2_orbit_idx); reshape(eye(6), [36,1])], options);
plot(xout_L2(:,1), xout_L2(:,2), 'LineWidth',2)

cj_L1 = jacobi_constant(V_family_L1_Lyapunov(:,l1_orbit_idx), mu);
cj_L2 = jacobi_constant(V_family_L2_Lyapunov(:,l2_orbit_idx), mu);
delta_cj = cj_L2 - cj_L1;

hold off
legend("L1", "L2", "Moon", "L1 Lyapunov Orbit", "L2 Lyapunov Orbit")
xlabel('$$\hat{x}$$','Interpreter','Latex', 'FontSize',18)
ylabel('$$\hat{y}$$','Interpreter','Latex', 'FontSize',18)
grid on
axis equal
title("L1 and L2 Lyapunov Orbits (\DeltaC_j = " + abs(delta_cj) + ")")
```

Manifolds

```
n_crossings = 2;
L1_manifold_time = 10;
L2_manifold_time = 6;

L1_manifolds = stable_manifolds(tout_L1, xout_L1, mu, l1_pos,
L1_manifold_time, n_crossings);
L2_manifolds = unstable_manifolds(tout_L2, xout_L2, mu, l2_pos,
L2_manifold_time, n_crossings);

poincare_stored = [];
stable_options = odeset('RelTol', 1e-12, 'AbsTol', 1e-12, 'Events', @(t,y)
stable_event(t, y, mu, n_crossings));
options = odeset('RelTol', 1e-12, 'AbsTol', 1e-12);
options_event = odeset('RelTol', 1e-12, 'AbsTol', 1e-12, 'Events', @(t,y)
eventFn(t, y, mu));
unstable_options = odeset('RelTol', 1e-12, 'AbsTol', 1e-12, 'Events', @(t,y)
unstable_event(t, y, mu, n_crossings));

figure(2)
plot(xout_L1(:,1), xout_L1(:,2), 'black', 'LineWidth', 3)
hold on
plot(xout_L2(:,1), xout_L2(:,2), 'black', 'LineWidth', 3)
scatter(l2_pos(1), l2_pos(2), 'filled', 'red')
scatter(p2_pos(1), p2_pos(2), 'filled', ' black')

iteration = 1;

for k = 1:length(L1_manifolds)
    count = 0;
    [moon_stable_t, moon_stable_x] = ode113(@(t, state)CR3BP(state, mu), [0,
-L1_manifold_time], L1_manifolds(:,k), stable_options);
    stable_end_times(k) = moon_stable_t(end);
    stable_end_states(k, :) = moon_stable_x(end, :);
    plot(moon_stable_x(:,1), moon_stable_x(:,2), 'blue')
    iteration = iteration + 1;
end
grid on
axis equal
xlabel('$$\hat{x}$$','Interpreter','Latex', 'FontSize',18)
ylabel('$$\hat{y}$$','Interpreter','Latex', 'FontSize',18)

poincare_stable = poincare_stored;

poincare_stored = [];

iteration = 1;

for k = 1:length(L2_manifolds)
    count = 0;
    [moon_unstable_t, moon_unstable_x] = ode113(@(t, state)CR3BP(state, mu),
[0, L2_manifold_time], L2_manifolds(:,k), unstable_options);
```

```
    unstable_end_times(k) = moon_unstable_t(end);
    unstable_end_states(k, :) = moon_unstable_x(end, :);
    if (abs(unstable_end_states(k, 1) - (1-mu)) < 1e-6 &&
unstable_end_states(k, 2) > 0)
        plot(moon_unstable_x(:,1), moon_unstable_x(:,2), 'red')
    end
    iteration = iteration + 1;
end
poincare_unstable = poincare_stored;

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

Poincare Map

```
figure(3)
subplot(1,2,1)
scatter(poincare_unstable(:,2), poincare_unstable(:,1), 10, 'filled', 'red');
hold on
scatter(poincare_stable(:,2), poincare_stable(:,1), 10, 'filled', 'blue');
xlabel("$\dot{y}$", 'Interpreter','latex')
ylabel("y")
title("Poincar\'e Map", 'Interpreter','latex')
grid on

stable_y_value = 0.0439618;
stable_ydot_value = 0.00816415;
[pc_stable_min, pc_stable_min_ind] = min(abs(poincare_stable(:,1) -
stable_y_value));
stable_ig = poincare_stable(pc_stable_min_ind, 3);

unstable_y_value = 0.0437366;
unstable_ydot_value = 0.00939577;
[pc_unstable_min, pc_unstable_min_ind] = min(abs(poincare_unstable(:,1) -
unstable_y_value));
unstable_ig = poincare_unstable(pc_unstable_min_ind, 3);

unstable_y_value_2 = 0.0441415;
unstable_ydot_value_2 = -0.155022;
[pc_unstable_min_2, pc_unstable_min_ind_2] = min(abs(poincare_unstable(:,1) -
unstable_y_value_2));
unstable_ig_2 = poincare_unstable(pc_unstable_min_ind_2, 3);

scatter(poincare_stable(pc_stable_min_ind, 2),
poincare_stable(pc_stable_min_ind, 1), 'filled', 'cyan')
scatter(poincare_unstable(pc_unstable_min_ind, 2),
poincare_unstable(pc_unstable_min_ind, 1), 'filled', 'black')
scatter(poincare_unstable(pc_unstable_min_ind_2, 2),
poincare_unstable(pc_unstable_min_ind_2, 1), 'filled', 'green')
legend("Unstable", "Stable", "Initial Guess - Stable", "Initial Guess -
Unstable (2B)", "Initial Guess - Unstable (2A)")
hold off
```

```
subplot(1,2,2)
scatter(poincare_unstable(:,2), poincare_unstable(:,1), 10, 'filled', 'red');
hold on
scatter(poincare_stable(:,2), poincare_stable(:,1), 10, 'filled', 'blue');
xlabel("$\dot{y}$", 'Interpreter','latex')
ylabel("y")
title("Poincar\'e Map (Zoomed in)", 'Interpreter','latex')
grid on
scatter(poincare_stable(pc_stable_min_ind, 2),
poincare_stable(pc_stable_min_ind, 1), 'filled', 'cyan')
scatter(poincare_unstable(pc_unstable_min_ind, 2),
poincare_unstable(pc_unstable_min_ind, 1), 'filled', 'black')
scatter(poincare_unstable(pc_unstable_min_ind_2, 2),
poincare_unstable(pc_unstable_min_ind_2, 1), 'filled', 'green')
legend("Unstable", "Stable", "Initial Guess - Stable", "Initial Guess -
Unstable (2B)", "Initial Guess - Unstable (2A)")
hold off
xlim([-0.2, 0.15])
ylim([0.036, 0.05])

sgtitle("Transfers 2A & 2B")
```

Multiple Shooting Initial Guess - 1

```
% Initial guess
x_1_0 = xout_L2(1, 1:6)';
delta_T1 = tout_L2(unstable_ig);
[tout, x_1] = ode113(@(t, state)CR3BP(state, mu), [0, delta_T1], x_1_0);
x_1_f = xout_L2(unstable_ig, 1:6)';

x_2_0 = L2_manifolds(:, unstable_ig);
delta_T2 = unstable_end_times(unstable_ig);
[tout, x_2] = ode113(@(t, state)CR3BP(state, mu), [0, delta_T2], x_2_0);
x_2_f = unstable_end_states(unstable_ig, :)';

x_3_f = L1_manifolds(:, stable_ig);
delta_T3 = -stable_end_times(stable_ig);
x_3_0 = stable_end_states(stable_ig, :)';

x_4_0 = xout_L1(stable_ig, 1:6)';
delta_T4 = V_family_L1_Lyapunov(7, l1_orbit_idx) - tout_L1(stable_ig);
[tout, x_4] = ode113(@(t, state)CR3BP(state, mu), [0, delta_T4], x_4_0);
x_4_f = x_4(end, 1:6)';

r_des_4 = xout_L1(1, 1:3)';

V0 = [x_1_0; delta_T1; x_2_0; delta_T2; x_3_0; delta_T3; x_4_0; delta_T4];

e1 = x_1_f - x_1_0;
e2 = x_2_f - x_2_0;
e3 = x_3_f - x_3_0;
```

```
V_soln = multiple_shooting(V0, mu, true, r_des_4);

flight_time_2B = V_soln(7) + V_soln(14) + V_soln(21) + V_soln(28);

% x_1_0 = xout_L1(unstable_ig, 1:6)';
% ig_um = L1_manifolds(:, unstable_ig);
% ig_sm = L2_manifolds(:, stable_ig);
% ig_L2_PO = xout_L2(stable_ig, 1:6)';
```

Delta V Calc

```
v_1_f = x_1_f(4:6);
v_2_0 = x_2_0(4:6);
v_2_f = x_2_f(4:6);
v_3_0 = x_3_0(4:6);
v_3_f = x_3_f(4:6);
v_4_0 = x_4_0(4:6);

dv1 = calc_dv(v_1_f, v_2_0);
dv2 = calc_dv(v_2_f, v_3_0);
dv3 = calc_dv(v_3_f, v_4_0);

dv_tot_2B = dv1 + dv2 + dv3;
dv_tot_dim_2B = dv_tot_2B * l_star_em / t_star_em;
```

Uncorrected Trajectory

```
[~, x_1] = ode113(@(t, state)CR3BP(state, mu), [0, V0(7)], V0(1:6), options);
[~, x_2] = ode113(@(t, state)CR3BP(state, mu), [0, V0(14)], V0(8:13),
options);
[~, x_3] = ode113(@(t, state)CR3BP(state, mu), [0, V0(21)], V0(15:20),
options);
[~, x_4] = ode113(@(t, state)CR3BP(state, mu), [0, V0(28)], V0(22:27),
options);

figure(5)
subplot(2, 1, 2)
scatter(l1_pos(1), l1_pos(2), 'filled', 'blue')
hold on
scatter(l2_pos(1), l2_pos(2), 'filled', 'red')
scatter(p2_pos(1), p2_pos(2), 'filled', 'black')

plot(x_1(:,1), x_1(:,2), 'LineWidth', 2)
plot(x_2(:,1), x_2(:,2), 'LineWidth', 2)
plot(x_3(:,1), x_3(:,2), 'LineWidth', 2)
plot(x_4(:,1), x_4(:,2), 'LineWidth', 2)

legend("L1", "L2", "Moon")
xlabel('$$\hat{x}$$','Interpreter','Latex', 'FontSize',18)
ylabel('$$\hat{y}$$','Interpreter','Latex', 'FontSize',18)
grid on
axis equal
title("2B")
```

```
sgtitle("2A and 2B Initial Guesses (Uncorrected) Transfers")
```

Corrected Trajectory

```
[~, x_1] = ode113(@(t, state)CR3BP(state, mu), [0, V_soln(7)], V_soln(1:6),
options);
[~, x_2] = ode113(@(t, state)CR3BP(state, mu), [0, V_soln(14)], V_soln(8:13),
options);
[~, x_3] = ode113(@(t, state)CR3BP(state, mu), [0, V_soln(21)],
V_soln(15:20), options);
[~, x_4] = ode113(@(t, state)CR3BP(state, mu), [0, V_soln(28)],
V_soln(22:27), options);

x1f = x_1(end,:);

figure(6)
subplot(2,1,2)
scatter(l1_pos(1), l1_pos(2), 'filled', 'blue')
hold on
scatter(l2_pos(1), l2_pos(2), 'filled', 'red')
scatter(p2_pos(1), p2_pos(2), 'filled', 'black')

plot(x_1(:,1), x_1(:,2), 'LineWidth', 2)
plot(x_2(:,1), x_2(:,2), 'LineWidth', 2)
plot(x_3(:,1), x_3(:,2), 'LineWidth', 2)
plot(x_4(:,1), x_4(:,2), 'LineWidth', 2)
plot(xout_L1(:,1), xout_L1(:,2), 'black', 'LineWidth', 2)
plot(xout_L2(:,1), xout_L2(:,2), 'black', 'LineWidth', 2)

hold off
legend("L1", "L2", "Moon")
xlabel('$$\hat{x}$$','Interpreter','Latex', 'FontSize',18)
ylabel('$$\hat{y}$$','Interpreter','Latex', 'FontSize',18)
grid on
axis equal
title("2B")

sgtitle("2A and 2B Corrected Transfers")
```

Multiple Shooting Initial Guess - 2

```
% Initial guess
x_1_0 = xout_L2(1, 1:6)';
delta_T1 = tout_L2(unstable_ig_2);
[tout, x_1] = ode113(@(t, state)CR3BP(state, mu), [0, delta_T1], x_1_0);
x_1_f = xout_L2(unstable_ig_2, 1:6)';

x_2_0 = L2_manifolds(:, unstable_ig_2);
delta_T2 = unstable_end_times(unstable_ig);
[tout, x_2] = ode113(@(t, state)CR3BP(state, mu), [0, delta_T2], x_2_0,
options_event);
delta_T2 = tout(end);
```

```
x_2_f = unstable_end_states(unstable_ig_2, :)';

x_3_f = L1_manifolds(:, stable_ig);
delta_T3 = -stable_end_times(stable_ig);
x_3_0 = stable_end_states(stable_ig, :)';

x_4_0 = xout_L1(stable_ig, 1:6)';
delta_T4 = V_family_L1_Lyapunov(7, l1_orbit_idx) - tout_L1(stable_ig);
[tout, x_4] = ode113(@(t, state)CR3BP(state, mu), [0, delta_T4], x_4_0);
x_4_f = x_4(end, 1:6)';

r_des_4 = xout_L1(1, 1:3)';

V0 = [x_1_0; delta_T1; x_2_0; delta_T2; x_3_0; delta_T3; x_4_0; delta_T4];

e1 = x_1_f - x_1_0;
e2 = x_2_f - x_2_0;
e3 = x_3_f - x_3_0;

V_soln = multiple_shooting(V0, mu, true, r_des_4);

flight_time_2A = V_soln(7) + V_soln(14) + V_soln(21) + V_soln(28);

v_1_f = x_1_f(4:6);
v_2_0 = x_2_0(4:6);
v_2_f = x_2_f(4:6);
v_3_0 = x_3_0(4:6);
v_3_f = x_3_f(4:6);
v_4_0 = x_4_0(4:6);

dv1 = calc_dv(v_1_f, v_2_0);
dv2 = calc_dv(v_2_f, v_3_0);
dv3 = calc_dv(v_3_f, v_4_0);

dv_tot_2A = dv1 + dv2 + dv3;
dv_tot_dim_2A = dv_tot_2A * l_star_em / t_star_em;

% x_1_0 = xout_L1(unstable_ig, 1:6)';
% ig_um = L1_manifolds(:, unstable_ig);
% ig_sm = L2_manifolds(:, stable_ig);
% ig_L2_PO = xout_L2(stable_ig, 1:6)';
```

Uncorrected Trajectory

```
[~, x_1] = ode113(@(t, state)CR3BP(state, mu), [0, V0(7)], V0(1:6), options);
[~, x_2] = ode113(@(t, state)CR3BP(state, mu), [0, V0(14)], V0(8:13),
options);
[~, x_3] = ode113(@(t, state)CR3BP(state, mu), [0, V0(21)], V0(15:20),
options);
[~, x_4] = ode113(@(t, state)CR3BP(state, mu), [0, V0(28)], V0(22:27),
options);
```

```
figure(5)
subplot(2, 1, 1)
scatter(l1_pos(1), l1_pos(2), 'filled', 'blue')
hold on
scatter(l2_pos(1), l2_pos(2), 'filled', 'red')
scatter(p2_pos(1), p2_pos(2), 'filled', 'black')

plot(x_1(:,1), x_1(:,2), 'LineWidth', 2)
plot(x_2(:,1), x_2(:,2), 'LineWidth', 2)
plot(x_3(:,1), x_3(:,2), 'LineWidth', 2)
plot(x_4(:,1), x_4(:,2), 'LineWidth', 2)

hold off
legend("L1", "L2", "Moon")
xlabel('$$\hat{x}$$','Interpreter','Latex', 'FontSize',18)
ylabel('$$\hat{y}$$','Interpreter','Latex', 'FontSize',18)
grid on
axis equal
title("2A")
```

Corrected Trajectory

```
[~, x_1] = ode113(@(t, state)CR3BP(state, mu), [0, V_soln(7)], V_soln(1:6),
options);
[~, x_2] = ode113(@(t, state)CR3BP(state, mu), [0, V_soln(14)], V_soln(8:13),
options);
[~, x_3] = ode113(@(t, state)CR3BP(state, mu), [0, V_soln(21)],
V_soln(15:20), options);
[~, x_4] = ode113(@(t, state)CR3BP(state, mu), [0, V_soln(28)],
V_soln(22:27), options);

x1f = x_1(end,:);
figure(6)
subplot(2,1,1)
scatter(l1_pos(1), l1_pos(2), 'filled', 'blue')
hold on
scatter(l2_pos(1), l2_pos(2), 'filled', 'red')
scatter(p2_pos(1), p2_pos(2), 'filled', 'black')

plot(x_1(:,1), x_1(:,2), 'LineWidth', 2)
plot(x_2(:,1), x_2(:,2), 'LineWidth', 2)
plot(x_3(:,1), x_3(:,2), 'LineWidth', 2)
plot(x_4(:,1), x_4(:,2), 'LineWidth', 2)
plot(xout_L1(:,1), xout_L1(:,2), 'black', 'LineWidth', 2)
plot(xout_L2(:,1), xout_L2(:,2), 'black', 'LineWidth', 2)

hold off
legend("L1", "L2", "Moon")
xlabel('$$\hat{x}$$','Interpreter','Latex', 'FontSize',18)
ylabel('$$\hat{y}$$','Interpreter','Latex', 'FontSize',18)
grid on
axis equal
title("2A")
```

```
function V_soln = multiple_shooting(V0, system_params, plot_input, r_des_4)
    % Script to compute a general three-dimensional periodic orbit via
multiple shooting
    % Inputs
    % V0 - initial guess for a free variable vector
    % statef_V0 - final state when V0 is used as initial guess using CR3BRP
    % EOMs
    % system_params - system parameters
    % r_des_4 - Desired position vector after 4th arc
    %
    % Output
    % V_soln - free variable vector corresponding to a solution

    % Get mass ratio of system
    mu = system_params(1);

    % Set tolerance for numerical integrator and constraint vector
    TOL = 1e-12;

    % Set options for ode113
    options = odeset('RelTol', TOL, 'AbsTol', TOL);

    % Propagate V0 non-linear CR3BP EOMs
    [tout, x1] = ode113(@(t, state)CR3BP(state, mu), [0 V0(7)], V0(1:6),
options);
    [tout, x2] = ode113(@(t, state)CR3BP(state, mu), [0 V0(14)], V0(8:13),
options);
    [tout, x3] = ode113(@(t, state)CR3BP(state, mu), [0 V0(21)], V0(15:20),
options);
    [tout, x4] = ode113(@(t, state)CR3BP(state, mu), [0 V0(28)], V0(22:27),
options);

    % Final final variables using V0
    % statef_V0 = xout(end,:);
    statef_x1 = x1(end,:)';
    statef_x2 = x2(end,:)';
    statef_x3 = x3(end,:)';
    statef_x4 = x4(end,:)';
    statef_V0 = [statef_x1; V0(7); statef_x2; V0(14); statef_x3; V0(21);
statef_x4; V0(28)];

    % Period is a free variable
    % T = V0(end);

    % Initialize constraint vector norm
    F_norm(1) = norm(F(V0, statef_V0, r_des_4));

    % Matrix of all free variable vectors
    V(:,1) = V0;

    % While loop params
    counter = 1;
    counter_max = 50;
    phi0 = reshape(eye(6), [36, 1]); % Initial phi is identity

    TOL = 1e-10;

    % While loop to reduce F_norm
    while ((F_norm(counter) > TOL) && (counter < counter_max))
        % x1_0 = [V(1:6,counter); phi0];
        % x2_0 = [V(8:13,counter); phi0];
        % x3_0 = [V(15:20,counter); phi0];
        % x4_0 = [V(22:27,counter); phi0];

        % Propagate full state and STM
        [~, x1_out] = ode113(@(t, state)CR3BP(state, mu), [0 V(7,counter)],
V(1:6,counter), options);
        [~, x2_out] = ode113(@(t,state)CR3BP(state, mu), [0 V(14,counter)],
V(8:13,counter), options);
        [~, x3_out] = ode113(@(t,state)CR3BP(state, mu), [0 V(21,counter)],
V(15:20,counter), options);
        [~, x4_out] = ode113(@(t,state)CR3BP(state, mu), [0 V(28,counter)],
V(22:27,counter), options);

        x1_f = x1_out(end, :)';
        x2_f = x2_out(end, :)';
        x3_f = x3_out(end, :)';
        x4_f = x4_out(end, :)';
        statef = [x1_f; V(7,counter); x2_f; V(14,counter); x3_f;
V(21,counter); x4_f; V(28,counter)];

        F_i = F(V(:,counter), statef, r_des_4);
        DF_i = DF_mat(V(:,counter), options, mu);

        % Find V_i+1
        V(:,counter+1) = V(:,counter) - DF_i' * inv(DF_i * DF_i') * F_i;

        % Calculate F_norm and update counter
        F_norm(counter+1) = norm(F_i);
        counter = counter + 1;
    end

    if plot_input
        figure()
        plot([1:counter], F_norm, '-o', 'LineWidth', 2)
        yscale log
        grid on
        xlabel("Iterations")
        ylabel("F Norm")
        title("Constraint Vector Norm for each Iteration")
        hold on
        tol_yline = ones([counter,1])*TOL;

        plot([1:counter], tol_yline, 'red', 'LineWidth', 2)
        hold off
        legend("Norm", "Threshold")
    end

    V_soln = V(:,end);

end

function out = DF_mat(V, options, mu)
    % Modified constraint DF matrix

    phi0 = reshape(eye(6), [36, 1]); % Initial phi is identity

    x1_0 = [V(1:6); phi0];
    x2_0 = [V(8:13); phi0];
    x3_0 = [V(15:20); phi0];
    x4_0 = [V(22:27); phi0];

    [~, x1_out] = ode113(@(t,state)CR3BP_full(state, mu), [0 V(7)], x1_0,
options);
    [~, x2_out] = ode113(@(t,state)CR3BP_full(state, mu), [0 V(14)], x2_0,
options);
    [~, x3_out] = ode113(@(t,state)CR3BP_full(state, mu), [0 V(21)], x3_0,
options);
    [~, x4_out] = ode113(@(t,state)CR3BP_full(state, mu), [0 V(28)], x4_0,
options);

    x1_f = x1_out(end, :)';
    x2_f = x2_out(end, :)';
    x3_f = x3_out(end, :)';
    x4_f = x4_out(end, :)';

    phi_row_1 = x1_f(7:end);
    phi_mat_1 = reshape(phi_row_1, [6,6])';
    phi_row_2 = x2_f(7:end);
    phi_mat_2 = reshape(phi_row_2, [6,6])';
    phi_row_3 = x3_f(7:end);
    phi_mat_3 = reshape(phi_row_3, [6,6])';
    phi_row_4 = x4_f(7:end);
    phi_mat_4 = reshape(phi_row_4, [6,6])';

    out = [phi_mat_1(1:3,:), x1_f(4:6), -eye([3,6]), zeros([3,15]);
            zeros([3,7]), phi_mat_2(1:3,:), x2_f(4:6), -eye([3,6]),
zeros([3,8]);
            zeros([3,14]), phi_mat_3(1:3,:), x3_f(4:6), -eye([3,6]),
zeros([3,1]);
            zeros([3,21]), phi_mat_4(1:3,:), x4_f(4:6)];

end

function out = F(state0, statef, r_des_4)
    % Modified Constraint Vector
    out = [statef(1:3) - state0(8:10);
            statef(8:10) - state0(15:17);
            statef(15:17) - state0(22:24);
            statef(22:24) - r_des_4];
```

```
end

function state_phi_dot = CR3BP_full(state_phi, mu)
    % Full state vector and state transition matrix differential equation
    % Inputs:
    % state_phi - Augmented state vector and STM [42x1]. The state vector -
    % [x0, y0, z0, x0_dot, y0_dot, z0_dot]. The STM - is 6x6 with each
    % element described as - phi_ij = dxi(tf)/dxj(t0). The phi matrix is
    % reshaped such that all the rows are concatenated vertically. For
    % example -
    % phi_mat = [phi11, phi12, phi13, ..., phi16;
    %            [phi21, phi22, phi23, ..., phi26;
    %            ...
    %            [phi61, phi62, phi63, ..., phi66]
    % becomes
    % phi_row = [phi11, phi12, ..., phi16, phi21, phi22, ..., phi66]'
    %
    % mu - system mass ratio [-]
    %
    % Output
    % state_phi_dot - Augmented state vector dot and STM_dot [42x1]. The
    % augmentation and reshaping scheme remains the same as the input.

    x = state_phi(1);
    y = state_phi(2);
    z = state_phi(3);
    xdot = state_phi(4);
    ydot = state_phi(5);
    zdot = state_phi(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);

    % Calc pseudo-potentials
    uxx = u_xx(mu, [x, y, z]);
    uyy = u_yy(mu, [x, y, z]);
    uxy = u_xy(mu, [x, y, z]);
    uzz = u_zz(mu, [x, y, z]);
    uxz = u_xz(mu, [x, y, z]);
    uyz = u_yz(mu, [x, y, z]);

    U_mat = [uxx, uxy uxz; uxy, uyy uyz; uxz uyz uzz];
    Omega = [0 2 0; -2 0 0; 0 0 0];
    A = [zeros(3), eye(3);
        U_mat, Omega];
    % Get only the phi elements into a row
    phi_row = state_phi(7:end);

    % Converting phi to matrix
    phi_mat = reshape(phi_row, [6,6])';

    % Get phi_dot
    phi_dot_mat = A * phi_mat;

    % Convert back to row
    phi_dot_row = reshape(phi_dot_mat', [36,1]);

    % Augment state and phi (in row form)
    state_phi_dot = [state_dot; phi_dot_row];

end
```

Published with MATLAB® R2024a