

---

## Table of Contents

.....	1
System Parameters(Earth-Moon): .....	1
Jacobi Synodic Frame Plot: .....	1
Generate Periodic Non-Planar Orbit About L1: .....	2
Manifolds: .....	3
Generate Lyapunov Orbit About L1: .....	3
Orbits Generated: .....	4
Generate Twin Orbits: .....	7
Generate Transfer Orbit Between Twin Orbits: .....	8
Simulate Elliptical Three-Body Problem and Compare to Circular Three Body Problem: .....	10
Simulating Transfer Orbit Response under Elliptical Case: .....	11
Equations of Motion for Circular Restricted Three Body Problem: .....	12
Equations of Motion for Elliptical Restricted Three Body Problem: .....	13
Hill Limiting Curve Generation: .....	13
State Transition Matrix Derivation: .....	13
Initial Conditions for Determining out of Plane Periodic Orbits, from Richardson's Paper: .....	14
Equations of Motion for CR3BP and State Transition Matrix Propagation: .....	16
Shooting Method For Determining Periodic Orbits: .....	17
Invariant Manifold Generation: .....	18
Poincare Map: .....	19
Generate Twin Orbits: .....	19
Generate Transfer Orbits: .....	20

`function CR3BP`

## System Parameters(Earth-Moon):

```
m1 = 5.9722*10^24; %Earth Mass(kg)
m2 = 7.34767*10^22; %Moon Mass(kg)

mu = (m1/m2+1)^(-1); %Jacobi Mu Non-Dimensional Constant

e = .0549; % Eccentricity of Earth-Moon System

% Lagrange Points of Earth Moon System:
L1 = [.83692,0];
L2 = [1.15568,0];
L3 = [-1.00506,0];
L4 = [.48785,.86603];
L5 = [.48785,-.86603];
```

## Jacobi Synodic Frame Plot:

```
p0 = [.6;0;.2]; v0 =[0;.4;0];
x0 = [p0;v0]; x0 = [-0.1, 0, 0, 0, -4.5500005869147468473556727985851,
    0]'; P = 6.2633709131030128602901640988421;
tspan = [0 P];
```

---

```

[t,x] =
    ode45(@(t,x)JacobiEOM(t,x,mu),tspan,x0,odeset('AbsTol',1e-12,'RelTol',1e-9));

C = @(x) 2*((1-mu)./sqrt((x(:,1)+mu).^2+x(:,2).^2+x(:,3).^2)+mu./
sqrt((x(:,1)-1+mu).^2+x(:,2).^2+x(:,3).^2))+x(:,1).^2+x(:,2).^2-
(x(:,4).^2+x(:,5).^2+x(:,6).^2);
E = -C(x);

[X,Y,CC] = HillCurve(mu,[-1.4 1.4],[-1.4 1.4]);

figure(1);
subplot(1,2,1); hold on; axis equal; grid on; axis([-1.4 1.4 -1.4
1.4]);
xlabel('$\xi$', 'interpreter', 'latex'); ylabel('$\eta$', 'interpreter', 'latex'); %%title('Circular-Planar Synodic Frame
Three-Body Problem', 'interpreter', 'latex', 'fontsize', 20);
plot([-1.4 1.4],[0 0], '--k');
plot([0 0],[-1.4 1.4], '--k');
plot(-mu,0, 'k.', 'MarkerSize', 40);
plot(1-mu,0, 'k.', 'MarkerSize', 20);
plot(0,0, 'k. ');

plot(L1(1),L1(2), 'b*');
plot(L2(1),L2(2), 'b*');
plot(L3(1),L3(2), 'b*');
plot(L4(1),L4(2), 'b*');
plot(L5(1),L5(2), 'b*');

xp = plot3(x(:,1),x(:,2),x(:,3));

contour(X,Y,CC,[-E(1) -E(1)]);

subplot(1,2,2); hold on;
xlabel('Time', 'interpreter', 'latex'); ylabel('Jacobi
Integral', 'interpreter', 'latex'); %%title('Jacobi
Integral', 'interpreter', 'latex', 'fontsize', 35);
xE = plot(t,E);

```

## Generate Periodic Non-Planar Orbit About L1:

```

Az = .08; % Approximate Z-Amplitude of Periodic Orbit
L = 2;
Lxi = L2(1);
NS = -1; % 1 for North, -1 for South
[x0,P] = RichardsonICs(Az,mu,L,Lxi,NS);

[XT,P] = ShootingMethod(x0,P,mu,[0;0;0],[3,5],[2,4,6]);
x0 = XT{end}(1,1:6)';

tspan = [0 P];
[t,x] =
    ode45(@(t,x)JacobiEOM(t,x,mu),tspan,x0,odeset('AbsTol',1e-12,'RelTol',1e-9));

```

---

---

```

E = -C(x);

figure(2); hold on; axis equal; grid on; axis([-1.4 1.7 -2.8 2.8]);
xlabel('$\xi$', 'interpreter', 'latex'); ylabel('$\eta$', 'interpreter', 'latex'); %%title('Circular-Planar Synodic Frame
    Three-Body Problem', 'interpreter', 'latex', 'fontsize', 20);
plot([-1.4 1.4], [0 0], '--k');
plot([0 0], [-1.4 1.4], '--k');
plot(-mu, 0, 'k.', 'MarkerSize', 40);
plot(1-mu, 0, 'k.', 'MarkerSize', 20);
plot(0, 0, 'k. ');

plot(L1(1), L1(2), 'b*');
plot(L2(1), L2(2), 'b*');
plot(L3(1), L3(2), 'b*');
plot(L4(1), L4(2), 'b*');
plot(L5(1), L5(2), 'b*');

plot3(x(:,1), x(:,2), x(:,3));

contour(X, Y, CC, [-E(1) -E(1)]);

```

## Manifolds:

```

n = 51; % Number Of Manifolds
[XSp, XUSp, XSm, XUSm] = Manifolds(x0, P, mu, n, 10^(-6), 3);

figure(2); hold on;
for i = 1:n
    plot3(XSp{i}(:,1), XSp{i}(:,2), XSp{i}(:,3), 'g');
    plot3(XSm{i}(:,1), XSm{i}(:,2), XSm{i}(:,3), 'g');
    plot3(XUSp{i}(:,1), XUSp{i}(:,2), XUSp{i}(:,3), 'r');
    plot3(XUSm{i}(:,1), XUSm{i}(:,2), XUSm{i}(:,3), 'r');
end

```

## Generate Lyapunov Orbit About L1:

```

x0 = [0, 0, 0, 0, 12.6841, 0]'; P = 6.52;

[XT, P] = ShootingMethod(x0, P, mu, [0;0], [5], [2,4]);
x0 = XT{end}(1,1:6)';

tspan = [0 P];
[t, x] =
    ode45(@(t,x) JacobiEOM(t,x,mu), tspan, x0, odeset('AbsTol', 1e-12, 'RelTol', 1e-9));

E = -C(x);

vpa(x0')
vpa(P)

```

---

---

# Orbits Generated:

L1 Orbits:

```
x0L1(1,:) = [0, 0, 0, 0, 12.684127017694370564981909410562, 0]; PL1(1)
= 6.5199256719550309213673244812526;
x0L1(2,:) = [0.1, 0, 0, 0, 3.9298123034825795940605530631728, 0];
PL1(2) = 7.0129906775240709038143904763274;
x0L1(3,:) = [0.2, 0, 0, 0, 2.6325619176619352579393762425752, 0];
PL1(3) = 7.2617352448394454356161986652296;
x0L1(4,:) = [0.3, 0, 0, 0, 1.9594195292790796969484290457331, 0];
PL1(4) = 7.4076834293913398354902710707393;
x0L1(5,:) = [0.4, 0, 0, 0, 1.504331533334239536355880773044, 0];
PL1(5) = 7.4488124224949379126314852328505;
x0L1(6,:) = [0.5, 0, 0, 0, 1.1571764079723902973739768640371, 0];
PL1(6) = 7.3330596158055332267622361541726;
x0L1(7,:) = [0.55, 0, 0, 0, 1.0095474950467668850961899806862, 0];
PL1(7) = 7.1781012276994191623202823393513;
x0L1(8,:) = [0.6, 0, 0, 0, 0.87514845577235378382141561814933, 0];
PL1(8) = 6.917873852929330880101588263642;
x0L1(9,:) = [0.65, 0, 0, 0, 0.75207717198377221912153345328989, 0];
PL1(9) = 6.498539831802730759591213427484;
x0L1(10,:) = [0.7, 0, 0, 0, 0.63837434660344627435080155919422, 0];
PL1(10) = 5.8385753576271781639661639928818;
x0L1(11,:) = [0.725, 0, 0, 0, 0.58364438876911650666556852229405, 0];
PL1(11) = 5.3853034163710571391447956557386;
x0L1(12,:) = [0.75, 0, 0, 0, 0.52785247220501829712446806297521, 0];
PL1(12) = 4.8296110191462524596772709628567;
x0L1(13,:) = [0.775, 0, 0, 0, 0.46393090429498928806140156666515, 0];
PL1(13) = 4.1517871460414301765240452368744;
x0L1(14,:) = [0.8, 0, 0, 0, 0.3573787201012048053705427719251, 0];
PL1(14) = 3.322541318963345258197250586818;
x0L1(15,:) = [0.81, 0, 0, 0, 0.27210571337190192497956786610303, 0];
PL1(15) = 2.9880672042687641543068366445368;
x0L1(16,:) = [0.82, 0, 0, 0, 0.16232232205367494448644549720484, 0];
PL1(16) = 2.7799088291599036715240345074562;
x0L1(17,:) = [0.83, 0, 0, 0, 0.060969504524787214438941163052732, 0];
PL1(17) = 2.7028849392929634731785881740507;
```

% L2 Orbits:

```
x0L2(1,:) = [1.5, 0, 0, 0, -0.86512419383989802579293382223113, 0];
PL2(1) = 8.4995214936597456301115016685799;
x0L2(2,:) = [1.45, 0, 0, 0, -0.79317309591324691719904649289674, 0];
PL2(2) = 8.1246780117271342191997973714024;
x0L2(3,:) = [1.4, 0, 0, 0, -0.72120460221370452646993953749188, 0];
PL2(3) = 7.6485832929594508300397137645632;
x0L2(4,:) = [1.35, 0, 0, 0, -0.64919157695364637650214945097105, 0];
PL2(4) = 7.0309024870558474162862694356591;
x0L2(5,:) = [1.3, 0, 0, 0, -0.57626469689484161040837761902367, 0];
PL2(5) = 6.2185224914238030891056041582488;
x0L2(6,:) = [1.275, 0, 0, 0, -0.53819327776181635769603417429607, 0];
PL2(6) = 5.7188193178690935880581491801422;
```

---

```

x0L2(7,:) = [1.25, 0, 0, 0, -0.4964166515596717399461113018333, 0];
PL2(7) = 5.1415168081160089741388219408691;
x0L2(8,:) = [1.225, 0, 0, 0, -0.44235210512263617443906582593627, 0];
PL2(8) = 4.4628016362565778862858678621706;
x0L2(9,:) = [1.2, 0, 0, 0, -0.31918285976731919362237022141926, 0];
PL2(9) = 3.6688298120953630743201756558847;
x0L2(10,:) = [1.19, 0, 0, 0, -0.22840667376646162090025882207556, 0];
PL2(10) = 3.4819277977300457926901344762882;
x0L2(11,:) = [1.18, 0, 0, 0, -0.14925354428773154880083495754661, 0];
PL2(11) = 3.4115332826998674420337920309976;
x0L2(12,:) = [1.17, 0, 0, 0, -0.082647950281856091825360977054515, 0];
PL2(12) = 3.3837928307077924650059230771149;

% L3 Orbits:
x0L3(1,:) = [-0.1, 0, 0, 0, -4.5500005869147468473556727985851, 0];
PL3(1) = 6.2633709131030128602901640988421;
x0L3(2,:) = [-0.2, 0, 0, 0, -2.9004745605404465003118730237475, 0];
PL3(2) = 6.2488589700763590428778115892783;
x0L3(3,:) = [-0.3, 0, 0, 0, -2.1373972823947311461267872800818, 0];
PL3(3) = 6.2398455733916158294505294179544;
x0L3(4,:) = [-0.4, 0, 0, 0, -1.6392404368687727433950840350008, 0];
PL3(4) = 6.2333581888865428055623851832934;
x0L3(5,:) = [-0.5, 0, 0, 0, -1.2618728100491374277680733939633, 0];
PL3(5) = 6.2284448429722969464705784048419;
x0L3(6,:) = [-0.6, 0, 0, 0, -0.95135209135081499898234369538841, 0];
PL3(6) = 6.2246939771302161048538437171374;
x0L3(7,:) = [-0.7, 0, 0, 0, -0.68224834516478582990828272158979, 0];
PL3(7) = 6.2218996048130756904015470354352;
x0L3(8,:) = [-0.8, 0, 0, 0, -0.44073216708298623700557072879747, 0];
PL3(8) = 6.2199505707370832396918558515608;
x0L3(9,:) = [-0.85, 0, 0, 0, -0.3275963922685305451665271903039, 0];
PL3(9) = 6.2192727144127752936242359282915;
x0L3(10,:) = [-0.9, 0, 0, 0, -0.21851829210631812383525129916961, 0];
PL3(10) = 6.2187860442528037907550242380239;
x0L3(11,:) = [-0.925, 0, 0, 0, -0.16530951773197899434819646558026,
0]; PL3(11) = 6.2186134239270796086884729447775;
x0L3(12,:) = [-0.95, 0, 0, 0, -0.11290010679780536351213271473171, 0];
PL3(12) = 6.2184875940958006523828771605622;
x0L3(13,:) = [-0.96, 0, 0, 0, -0.092146580404271039022034983645426,
0]; PL3(13) = 6.2184503236397237202481846907176;
x0L3(14,:) = [-0.97, 0, 0, 0, -0.071508334802316581013670315769559,
0]; PL3(14) = 6.2184205048698828477427014149725;
x0L3(15,:) = [-0.98, 0, 0, 0, -0.050981153879406083950787120784298,
0]; PL3(15) = 6.2183981323792263395944246440195;

figure(3);
subplot(1,3,1); hold on;
plot(x0L1(:,1),x0L1(:,5),'Color',[0,0,1]);
plot(x0L2(:,1),x0L2(:,5),'Color',[0,1,1]);
plot(x0L3(:,1),x0L3(:,5),'Color',[1,0,1]);
plot([L1(1) L1(1)],ylim,'--k');
plot([L2(1) L2(1)],ylim,'--k');
plot([L3(1) L3(1)],ylim,'--k');

```

---

---

```

xlabel('$\xi$', 'Interpreter', 'latex', 'fontsize', 12); ylabel('$
\dot{\eta}$', 'Interpreter', 'latex', 'fontsize', 12);
%title('$\dot{\eta}$ vs $\xi$ for Lyapunov Orbits About L1, L2 and
L3', 'Interpreter', 'latex');

subplot(1,3,2); hold on;
plot(x0L1(:,1),C(x0L1), 'Color',[0,0,1]);
plot(x0L2(:,1),C(x0L2), 'Color',[0,1,1]);
plot(x0L3(:,1),C(x0L3), 'Color',[1,0,1]);
plot([L1(1) L1(1)],ylim,'--k');
plot([L2(1) L2(1)],ylim,'--k');
plot([L3(1) L3(1)],ylim,'--k');
xlabel('$\xi$', 'Interpreter', 'latex', 'fontsize', 12); ylabel('Jacobi
Integral', 'Interpreter', 'latex', 'fontsize', 12);
%title('C vs $\xi$ for Lyapunov Orbits About L1, L2 and
L3', 'Interpreter', 'latex');
legend('L1 Lyapunov Orbit', 'L2 Lyapunov Orbit', 'L3 Lyapunov
Orbit', 'L1', 'L2', 'L3', 'Interpreter', 'latex', 'Location', 'southoutside', 'fontsize'
8);

subplot(1,3,3); hold on;
plot(x0L1(:,1),PL1, 'Color',[0,0,1]);
plot(x0L2(:,1),PL2, 'Color',[0,1,1]);
plot(x0L3(:,1),PL3, 'Color',[1,0,1]);
plot([L1(1) L1(1)],ylim,'--k');
plot([L2(1) L2(1)],ylim,'--k');
plot([L3(1) L3(1)],ylim,'--k');
xlabel('$\xi$', 'Interpreter', 'latex', 'fontsize', 12); ylabel('Orbital
Period', 'Interpreter', 'latex', 'fontsize', 12);
%title('Orbital Period vs $\xi$ for Lyapunov Orbits About L1, L2 and
L3', 'Interpreter', 'latex');

figure(4); hold on; axis equal; grid on; axis([-2 2 -2 2]);
xlabel('$\xi$'); ylabel('$\eta$'); %title('Twin
Orbits', 'interpreter', 'latex', 'fontsize', 20);
plot([-2 2],[0 0], '--k');
plot([0 0],[-2 2], '--k');
plot(-mu,0, 'k.', 'MarkerSize', 40);
plot(1-mu,0, 'k.', 'MarkerSize', 20);
plot(0,0, 'k. ');
plot(L1(1),L1(2), 'r*');
plot(L2(1),L2(2), 'r*');
plot(L3(1),L3(2), 'r*');
plot(L4(1),L4(2), 'r*');
plot(L5(1),L5(2), 'r*');
for i = 1:length(PL1)
    [~,x] = ode45(@(t,x)JacobiEOM(t,x,mu),[0
    PL1(i)],x0L1(i,:),odeset('AbsTol',1e-12,'RelTol',1e-9));
    plot3(x(:,1),x(:,2),x(:,3), 'Color',[0,0,1]);
end
for i = 1:length(PL2)
    [~,x] = ode45(@(t,x)JacobiEOM(t,x,mu),[0
    PL2(i)],x0L2(i,:),odeset('AbsTol',1e-12,'RelTol',1e-9));
    plot3(x(:,1),x(:,2),x(:,3), 'Color',[0,1,1]);

```

---

---

```

end
for i = 1:length(PL3)
    [~,x] = ode45(@(t,x)JacobiEOM(t,x,mu),[0
    PL3(i)],x0L3(i,:),odeset('AbsTol',1e-12,'RelTol',1e-9));
    plot3(x(:,1),x(:,2),x(:,3),'Color',[1,0,1]);
end

```

## Generate Twin Orbits:

```

Cd = 3;
[x01,P1] = OrbitalEnergyTrack(Cd,x0L1,PL1,mu);
[x02,P2] = OrbitalEnergyTrack(Cd,x0L2,PL2,mu);
[x03,P3] = OrbitalEnergyTrack(Cd,x0L3,PL3,mu);

[t1,x1] = ode45(@(t,x)JacobiEOM(t,x,mu),[0
    P1],x01,odeset('AbsTol',1e-12,'RelTol',1e-9));
[t2,x2] = ode45(@(t,x)JacobiEOM(t,x,mu),[0
    P2],x02,odeset('AbsTol',1e-12,'RelTol',1e-9));
[t3,x3] = ode45(@(t,x)JacobiEOM(t,x,mu),[0
    P3],x03,odeset('AbsTol',1e-12,'RelTol',1e-9));

E1 = -C(x1); E2 = -C(x2); E3 = -C(x3);

[X,Y,CC] = HillCurve(mu,[-1.4 1.4],[-1.4 1.4]);

figure(5);
subplot(1,2,1); hold on; axis equal; grid on; axis([-1.4 1.4 -1.4
    1.4]);
xlabel('\xi'); ylabel('\eta'); %title('Twin
    Orbits','interpreter','latex','fontsize',20);
plot([-1.4 1.4],[0 0],'--k');
plot([0 0],[-1.4 1.4],'--k');
plot(-mu,0,'k.','MarkerSize',40);
plot(1-mu,0,'k.','MarkerSize',20);
plot(0,0,'k. ');
plot(L1(1),L1(2),'b*');
plot(L2(1),L2(2),'b*');
plot(L3(1),L3(2),'b*');
plot(L4(1),L4(2),'b*');
plot(L5(1),L5(2),'b*');
plot3(x1(:,1),x1(:,2),x1(:,3),'Color',[0,0,1]);
plot3(x2(:,1),x2(:,2),x2(:,3),'Color',[0,1,1]);
plot3(x3(:,1),x3(:,2),x3(:,3),'Color',[1,0,1]);
contour(X,Y,CC,[-E1(1) -E1(1)]);

subplot(1,2,2); hold on;
xlabel('Time','interpreter','latex'); ylabel('Jacobi
    Integral','interpreter','latex'); %title('Jacobi
    Integral','interpreter','latex','fontsize',35);
plot(t1,E1,'Color',[0,0,1]);
plot(t2,E2,'Color',[0,1,1]);
plot(t3,E3,'Color',[1,0,1]);

```

---

# Generate Transfer Orbit Between Twin Orbits:

L1 to L2:

```
Cd = 3.13;
xiP = 1.02-mu;
pert1 = 10^(-5); nP1 = 3;
pert2 = 10^(-5); nP2 = 3;
[xt,x01,x02,PMUS1,PMS2,P1,P2,P01,P02] =
    TransferOrbit(x0L1,x0L2,PL1,PL2,Cd,xiP,mu,pert1,nP1,pert2,nP2);
[t1,x1] = ode45(@(t,x)JacobiEOM(t,x,mu),[0
    P01],x01,odeset('AbsTol',1e-12,'RelTol',1e-9));
[t2,x2] = ode45(@(t,x)JacobiEOM(t,x,mu),[0
    P02],x02,odeset('AbsTol',1e-12,'RelTol',1e-9));
% % L2 to L1:
% Cd = 3.13;
% xiP = 1.02-mu;
% pert1 = 10^(-5); nP1 = 3;
% pert2 = 10^(-5); nP2 = 3;
% [xt,x01,x02,PMUS1,PMS2,P1,P2,P01,P02] =
%     TransferOrbit(x0L2,x0L1,PL2,PL1,Cd,xiP,mu,pert2,nP2,pert1,nP1);
% [t1,x1] = ode45(@(t,x)JacobiEOM(t,x,mu),[0
%     P01],x01,odeset('AbsTol',1e-12,'RelTol',1e-9));
% [t2,x2] = ode45(@(t,x)JacobiEOM(t,x,mu),[0
%     P02],x02,odeset('AbsTol',1e-12,'RelTol',1e-9));
% vpa(xt(1,:))
% %L1 to L3:
% Cd = 3;
% xiP = -.2;
% pert3 = 10^(-3); nP3 = 10;
% pert1 = 10^(-5); nP1 = 3;
% [xt,x01,x02,PMUS1,PMS2,P1,P2,P01,P02] =
%     TransferOrbit(x0L1,x0L3,PL1,PL3,Cd,xiP,mu,pert1,nP1,pert3,nP3);
% [t1,x1] = ode45(@(t,x)JacobiEOM(t,x,mu),[0
%     P01],x01,odeset('AbsTol',1e-12,'RelTol',1e-9));
% [t2,x2] = ode45(@(t,x)JacobiEOM(t,x,mu),[0
%     P02],x02,odeset('AbsTol',1e-12,'RelTol',1e-9));
% vpa(xt(1,:))
% %L3 to L1:
% Cd = 3;
% xiP = -.2;
% pert3 = 10^(-3); nP3 = 10;
% pert1 = 10^(-5); nP1 = 3;
% [xt,x01,x02,PMUS1,PMS2,P1,P2,P01,P02] =
%     TransferOrbit(x0L3,x0L1,PL3,PL1,Cd,xiP,mu,pert3,nP3,pert1,nP1);
% [t1,x1] = ode45(@(t,x)JacobiEOM(t,x,mu),[0
%     P01],x01,odeset('AbsTol',1e-12,'RelTol',1e-9));
% [t2,x2] = ode45(@(t,x)JacobiEOM(t,x,mu),[0
%     P02],x02,odeset('AbsTol',1e-12,'RelTol',1e-9));
% vpa(xt(1,:))
% %L2 to L3:
% Cd = 3;
% xiP = -.2;
```



---

```

% pert3 = 10^(-3); nP3 = 10;
% pert2 = 10^(-5); nP2 = 3;
% [xt,x01,x02,PMUS1,PMS2,P1,P2,P01,P02] =
    TransferOrbit(x0L2,x0L3,PL2,PL3,Cd,xiP,mu,pert2,nP2,pert3,nP3);
% [t1,x1] = ode45(@(t,x)JacobiEOM(t,x,mu),[0
    P01],x01,odeset('AbsTol',1e-12,'RelTol',1e-9));
% [t2,x2] = ode45(@(t,x)JacobiEOM(t,x,mu),[0
    P02],x02,odeset('AbsTol',1e-12,'RelTol',1e-9));
% vpa(xt(1,:))
% %L3 to L2:
% Cd = 3;
% xiP = -.2;
% pert3 = 10^(-3); nP3 = 10;
% pert2 = 10^(-5); nP2 = 3;
% [xt,x01,x02,PMUS1,PMS2,P1,P2,P01,P02] =
    TransferOrbit(x0L3,x0L2,PL3,PL2,Cd,xiP,mu,pert3,nP3,pert2,nP2);
% [t1,x1] = ode45(@(t,x)JacobiEOM(t,x,mu),[0
    P01],x01,odeset('AbsTol',1e-12,'RelTol',1e-9));
% [t2,x2] = ode45(@(t,x)JacobiEOM(t,x,mu),[0
    P02],x02,odeset('AbsTol',1e-12,'RelTol',1e-9));
% vpa(xt(1,:))

figure(6);
subplot(1,3,1); hold on; grid on; axis equal; axis([.7 1.3 -.3 .3]);
xlabel('$\xi$', 'interpreter', 'latex'); ylabel('$\eta$
$', 'interpreter', 'latex', 'fontweight', 'bold'); %title('Twin
    Orbits', 'interpreter', 'latex', 'fontsize', 20);
plot([-1.4 1.4],[0 0], '--k');
plot([0 0],[-1.4 1.4], '--k');
plot(-mu,0, 'k.', 'MarkerSize', 40);
plot(1-mu,0, 'k.', 'MarkerSize', 20);
plot(0,0, 'k. ');
plot(L1(1),L1(2), 'b*');
plot(L2(1),L2(2), 'b*');
plot(L3(1),L3(2), 'b*');
plot(L4(1),L4(2), 'b*');
plot(L5(1),L5(2), 'b*');
for i = 1:20:length(P1)
    if ~isempty(P1{i})
        plot3(P1{i}(:,1),P1{i}(:,2),P1{i}(:,3), 'r');
    end
    if ~isempty(P2{i})
        plot3(P2{i}(:,1),P2{i}(:,2),P2{i}(:,3), 'g');
    end
end
plot([xiP xiP],[-.2 .2], 'k', 'MarkerSize', 3);
contour(X,Y,CC,[Cd Cd]);

subplot(1,3,2); hold on; axis([-0.15 .1 -.6 1]);
plot(PMUS1(:,1),PMUS1(:,2), 'r');
plot(PMS2(:,1),PMS2(:,2), 'g');
xlabel('$\eta$', 'interpreter', 'latex', 'fontweight', 'bold'); ylabel('$
\dot{\eta}$', 'interpreter', 'latex', 'fontweight', 'bold');

```

---

---

```

subplot(1,3,3); hold on; axis equal; grid on; axis([.7 1.3 -.3 .3]);
xlabel('$\xi$', 'interpreter', 'latex', 'fontweight', 'bold'); ylabel('$\eta$', 'interpreter', 'latex', 'fontweight', 'bold'); %title('Twin
Orbits', 'interpreter', 'latex', 'fontsize', 20);
plot([-1.4 1.4],[0 0], '--k');
plot([0 0],[-1.4 1.4], '--k');
plot(-mu,0, 'k.', 'MarkerSize', 40);
plot(1-mu,0, 'k.', 'MarkerSize', 20);
plot(0,0, 'k. ');

plot(L1(1),L1(2), 'b*');
plot(L2(1),L2(2), 'b*');
plot(L3(1),L3(2), 'b*');
plot(L4(1),L4(2), 'b*');
plot(L5(1),L5(2), 'b*');

plot3(x1(:,1),x1(:,2),x1(:,3), 'g');
plot3(x2(:,1),x2(:,2),x2(:,3), 'r');
plot3(xt(:,1),xt(:,2),xt(:,3), 'b');

contour(X,Y,CC,[Cd Cd]);

```

## Simulate Elliptical Three-Body Problem and Compare to Circular Three Body Problem:

```

p0 = [.6;0;.2]; v0 =[0;.4;0]; f0 = 0;
x0 = [p0;v0;f0];
tspan = [0 100];

[tC,xC] =
ode45(@(t,x)JacobiEOM(t,x,mu),tspan,x0(1:6),odeset('AbsTol',1e-12,'RelTol',1e-9))
[tE,xE] =
ode45(@(t,x)ElipticalEOM(t,x,e,mu),tspan,x0,odeset('AbsTol',1e-12,'RelTol',1e-9))

C = @(x) 2*((1-mu)./sqrt((x(:,1)+mu).^2+x(:,2).^2+x(:,3).^2)+mu./
sqrt((x(:,1)-1+mu).^2+x(:,2).^2+x(:,3).^2))+x(:,1).^2+x(:,2).^2-
(x(:,4).^2+x(:,5).^2+x(:,6).^2);
EC = -C(xC);
EE = - C(xE);

figure(7);
subplot(2,3,[1 4]); hold on; axis equal; grid on; axis([-1.4 1.4 -1.4
1.4]);
xlabel('$\xi$', 'interpreter', 'latex'); ylabel('$\eta$', 'interpreter', 'latex'); %title('Circular-Planar Synodic Frame
Three-Body Problem', 'interpreter', 'latex', 'fontsize', 20);
plot([-1.4 1.4],[0 0], '--k');
plot([0 0],[-1.4 1.4], '--k');
plot(-mu,0, 'k.', 'MarkerSize', 40);
plot(1-mu,0, 'k.', 'MarkerSize', 20);
plot(0,0, 'k. ');

```

---

```

plot(L1(1),L1(2), 'b*');
plot(L2(1),L2(2), 'b*');
plot(L3(1),L3(2), 'b*');
plot(L4(1),L4(2), 'b*');
plot(L5(1),L5(2), 'b*');

plot3(xE(:,1),xE(:,2),xE(:,3));

subplot(2,3,[2 5]); hold on;
xlabel('Time','interpreter','latex'); ylabel('Jacobi
Integral','interpreter','latex');%%title('Jacobi
Integral','interpreter','latex','fontsize',35);
plot(tC,EC);
plot(tE,EE);

subplot(2,3,3); hold on;
xlabel('Time','interpreter','latex'); ylabel('Moon True
Anomaly[rad]','interpreter','latex');
plot(tE,xE(:,7));

subplot(2,3,6); hold on;
xlabel('Time','interpreter','latex'); ylabel('Elliptical-Circular State
Error','interpreter','latex');
maxt = min([length(tC) length(tE)]);
plot(tE(1:maxt),sqrt(sum((xC(1:maxt,:)-xE(1:maxt,1:6)).^2)));

```

## Simulating Transfer Orbit Response under Elliptical Case:

[illegible]



---

end

## Equations of Motion for Elliptical Restricted Three Body Problem:

```
function xdot = EllipticalEOM(t,x,e,mu)

rho13 = ((x(1)+mu)^2+x(2)^2+x(3)^2)^(3/2);
rho23 = ((x(1)+mu-1)^2+x(2)^2+x(3)^2)^(3/2);

xdot = [x(4)
        x(5)
        x(6)
        -(1-mu)*(x(1)+mu)/rho13-mu*(x(1)+mu-1)/rho23+x(1))/(1+e*cos(x(7)))+2*x(5)
        -(1-mu)/rho13+mu/rho23)*x(2)+x(2))/(1+e*cos(x(7)))-2*x(4)
        -(1-mu)/rho13+mu/rho23)*x(3)-e*cos(x(7))/(1+e*cos(x(7)))
        (1+e*cos(x(7)))^2/(1-e^2)^(3/2)];
```

end

## Hill Limiting Curve Generation:

```
function [X,Y,C] = HillCurve(mu,xlim,ylim)

[X,Y] =
    meshgrid(linspace(xlim(1),xlim(2),1000),linspace(ylim(1),ylim(2),1000));

r1 = sqrt((X+mu).^2+Y.^2);
r2 = sqrt((X-1+mu).^2+Y.^2);
C = 2*((1-mu)*(.5*r1.^2+r1.^(-1))+mu*(.5*r2.^2+r2.^(-1)));
```

end

## State Transition Matrix Derivation:

```
function STM

syms xi eta zeta xidot etadot zetadot mu
xidotdot = -(1-mu)*(xi+mu)/((xi+mu)^2+eta^2+zeta^2)^(3/2)-mu*(xi+mu-1)/((xi+mu-1)^2+eta^2+zeta^2)^(3/2)+2*etadot+xi;
etadotdot = -((1-mu)/((xi+mu)^2+eta^2+zeta^2)^(3/2)+mu/((xi+mu-1)^2+eta^2+zeta^2)^(3/2))*eta-2*xidot+eta;
zetadotdot = -((1-mu)/((xi+mu)^2+eta^2+zeta^2)^(3/2)+mu/((xi+mu-1)^2+eta^2+zeta^2)^(3/2))*zeta;

Uxx = [diff(xidotdot,xi) diff(xidotdot,eta) diff(xidotdot,zeta)
        diff(etadotdot,xi) diff(etadotdot,eta) diff(etadotdot,zeta)
        diff(zetadotdot,xi) diff(zetadotdot,eta)
        diff(zetadotdot,zeta)];
```

---

```

A = [zeros(3,3) eye(3,3)
      Uxx [0 2 0;-2 0 0;0 0 0]];

pretty(A)
disp('A is dependent on xi, eta, zeta, and mu');
end

```

## Initial Conditions for Determining out of Plane Periodic Orbits, from Richardson's Paper:

```

function [x0,P] = RichardsonICs(Az,mu,L,Lxi,NS)

if L == 1
    gamL = (1-mu)-Lxi;
    n = 1:4;
    c = 1/gamL^3*(mu+(-1).^(n).*(1-mu).*gamL.^(1+n)./(1-gamL).^(n+1));
    n = 2-NS;
elseif L == 2
    gamL = Lxi-(1-mu);
    n = 1:4;
    c = 1/gamL^3*((-1).^(n)*mu+(-1).^(n).*(1-mu).*gamL.^(1+n)./(1+gamL).^(n+1));
    n = 2+NS;
elseif L == 3
    gamL = (1-mu)-Lxi;
    n = 1:4;
    c = 1/gamL^3*((1-mu)+mu*gamL.^(1+n)./(1+gamL).^(n+1));
    n = 2-NS;
else
    display('Invalid Libration Point');
end

lam = sqrt((-c(2)-2)+sqrt((c(2)-2)^2+4*(c(2)-1)*(1+2*c(2))))/2);
k = 1/(2*lam)*(lam^2+1+2*c(2));

d1 = 3*lam^2/k*(k*(6*lam^2-1)-2*lam);
d2 = 8*lam^2/k*(k*(11*lam^2-1)-2*lam);

b21 = -3*c(3)*lam/(2*d1)*(3*k*lam-4);
b22 = 3*c(3)*lam/d1;

d21 = -c(3)/(2*lam^2);

a21 = 3*c(3)*(k^2-2)/(4*(1+2*c(2)));
a22 = 3*c(3)/(4*(1+2*c(2)));
a23 = -3*c(3)*lam/(4*k*d1)*(3*k^3*lam-6*k*(k-lam)+4);
a24 = -3*c(3)*lam/(4*k*d1)*(2+3*k*lam);
a31 = -9*lam/(4*d2)*(4*c(3)*(k*a23-b21)+k*c(4)*(4+k^2))+(9*lam^2+1-c(2))/(2*d2)*(3*c(3)*(2*a23-k*b21)+c(4)*(2+3*k^2));
a32 = -1/d2*(9*lam/4*(4*c(3)*(k*a24-b22)+k*c(4))+3/2*(9*lam^2+1-c(2))*(c(3)*(k*b22+d21-2*a24)-c(4)));

```

---

---

```

b31 = 3/(8*d2)*(8*lam*(3*c(3)*(k*b21-2*a23)-
c(4)*(2+3*k^2))+(9*lam^2+1+2*c(2))*(4*c(3)*(k*a23-
b21)+k*c(4)*(4+k^2)));
b32 = 1/d2*(9*lam*(c(3)*(k*b22+d21-2*a24)-
c(4))+3/8*(9*lam^2+1+2*c(2))*(4*c(3)*(k*a24-b22)+k*c(4)));

d31 = 3/(64*lam^2)*(4*c(3)*a24+c(4));
d32 = 3/(64*lam^2)*(4*c(3)*(a23-d21)+c(4)*(4+k^2));

a1 = -3/2*c(3)*(2*a21+a23+5*d21)-3/8*c(4)*(12-k^2);
a2 = 3/2*c(3)*(a24-2*a22)+9/8*c(4);

s1 = 1/(2*lam*(lam*(1+k^2)-2*k))*(3/2*c(3)*(2*a21*(k^2-2)-
a23*(k^2+2)-2*k*b21)-3/8*c(4)*(3*k^4-8*k^2+8));
s2 = 1/
(2*lam*(lam*(1+k^2)-2*k))*(3/2*c(3)*(2*a22*(k^2-2)+a24*(k^2+2)+2*k*b22+5*d21)+3/8*
k^2));

l1 = a1 + 2*lam^2*s1;
l2 = a2+2*lam^2*s2;

del = lam^2-c(2);
deln = (2-n); % 1 or 3

Az = Az/gamL;
Ax = sqrt((-del-l2*Az^2)/l1);

Om = 1+s1*Ax^2+s2*Az^2;

if L == 2
    xi0 = a21*Ax^2+a22*Az^2+Ax+(a23*Ax^2-a24*Az^2)-(a31*Ax^3-
a32*Ax*Az^2);
    eta0 = 0;
    zeta0 = -deln*Az+deln*d21*Ax*Az*(-2)-deln*(d32*Az*Ax^2-d31*Az^3);

    xidot0 = 0;
    etadot0 = lam*Om*(-k*Ax+2*(b21*Ax^2-b22*Az^2)-3*(b31*Ax^3-
b32*Ax*Az^2));
    zetadot0 = 0;
else
    xi0 = a21*Ax^2+a22*Az^2-Ax+(a23*Ax^2-a24*Az^2)+(a31*Ax^3-
a32*Ax*Az^2);
    eta0 = 0;
    zeta0 = deln*Az+deln*d21*Ax*Az*(-2)+deln*(d32*Az*Ax^2-d31*Az^3);

    xidot0 = 0;
    etadot0 = lam*Om*(k*Ax+2*(b21*Ax^2-b22*Az^2)+3*(b31*Ax^3-
b32*Ax*Az^2));
    zetadot0 = 0;
end

P = 2*pi/(lam*Om);

```

---

---

```

x0 = [xi0;eta0;zeta0;xidot0;etadot0;zetadot0]*gamL +
Lxi*[1;zeros(5,1)];

```

```

end

```

## Equations of Motion for CR3BP and State Transition Matrix Propagation:

```

function dxphi = XSTMEOM(t,xphi,mu)

x = xphi(1:6);
phi = reshape(xphi(7:end),6,6);

rho13 = ((x(1)+mu)^2+x(2)^2+x(3)^2)^(3/2);
rho23 = ((x(1)+mu-1)^2+x(2)^2+x(3)^2)^(3/2);
xdot = [x(4)
        x(5)
        x(6)
        -(1-mu)*(x(1)+mu)/rho13-mu*(x(1)+mu-1)/rho23+2*x(5)+x(1)
        -((1-mu)/rho13+mu/rho23)*x(2)-2*x(4)+x(2)
        -((1-mu)/rho13+mu/rho23)*x(3)];

rho1 = (x(1)+mu).^2+x(2).^2+x(3).^2;
rho2 = (x(1)-1+mu).^2+x(2).^2+x(3).^2;

Uxx = [(mu - 1)/rho1^(3/2) - mu/rho2^(3/2) + (3*mu*(2*mu + 2*x(1)
- 2)*(mu + x(1) - 1))/(2*rho2^(5/2)) - (3*(2*mu + 2*x(1))*(mu +
x(1))*(mu - 1))/(2*rho1^(5/2)) + 1,(3*x(2)*mu*(mu + x(1) - 1))/
rho2^(5/2) - (3*x(2)*(mu + x(1))*(mu - 1))/rho1^(5/2),(3*mu*x(3)*(mu +
x(1) - 1))/rho2^(5/2) - (3*x(3)*(mu + x(1))*(mu - 1))/rho1^(5/2)
-x(2)*((3*(2*mu + 2*x(1))*(mu - 1))/(2*rho1^(5/2)) -
(3*mu*(2*mu + 2*x(1) - 2))/(2*rho2^(5/2))), (mu - 1)/rho1^(3/2) -
mu/rho2^(3/2) + x(2)*((3*x(2)*mu)/rho2^(5/2) - (3*x(2)*(mu - 1))/
rho1^(5/2)) + 1,x(2)*((3*mu*x(3))/rho2^(5/2) - (3*x(3)*(mu - 1))/
rho1^(5/2))
-x(3)*((3*(2*mu + 2*x(1))*(mu - 1))/(2*rho1^(5/2)) -
(3*mu*(2*mu + 2*x(1) - 2))/(2*rho2^(5/2))),x(3)*((3*x(2)*mu)/
rho2^(5/2) - (3*x(2)*(mu - 1))/rho1^(5/2)), x(3)*((3*mu*x(3))/
rho2^(5/2) - (3*x(3)*(mu - 1))/rho1^(5/2)) + (mu - 1)/rho1^(3/2) - mu/
rho2^(3/2)];

A = [zeros(3,3),eye(3,3)
     Uxx,[0 2 0;-2 0 0;0 0 0]];

phidot = A*phi;

dxphi = [xdot; reshape(phidot,36,1)];

end

```

---



---

# Shooting Method For Determining Periodic Orbits:

```
function [XT,P,t] = ShootingMethod(x0,P,mu,xd,x0S,xfS)

tf = P/2;
maxiter = 100;
tol = 10^(-8);
i = 0;

%figure(1); hold on;

while 1

    tspan = linspace(0,tf,1000);
    xphi0 = [x0; reshape(eye(6,6),36,1)];
    [t,xphi] =
ode45(@(t,xphi)XSTMEOM(t,xphi,mu),tspan,xphi0,odeset('AbsTol',1e-12,'RelTol',1e-9)

    xf = xphi(end,1:6)';
    phif = reshape(xphi(end,7:end),6,6);

    xfdot = JacobiEOM(t,xf,mu);
    xe = xd-xf(xfS);

    if norm(xe)<tol || i>maxiter
        break;
    end

    K = [phif(xfS,x0S) xfdot(xfS)];
    Delta = K'*inv(K*K')*xe;

    x0(x0S) = x0(x0S)+Delta(1:end-1);
    tf = tf + Delta(end);

    i = i + 1;
    XT{i} = xphi(:,1:6);

end

if i>maxiter
    disp('Shoting Algorithm Did Not Converge Before Maximum Iteration
    Limit');
else
    disp(sprintf('Shoting Algorithm Converged in %d iterations',i));
end

P = tf*2;

end
```

---

# Invariant Manifold Generation:

```
function [XSp,XUSp,XSm,XUSm,tSp,tUSp,tSm,tUSm,to] =  
    Manifolds(x0,P,mu,n,pert,nP,trange)  
  
    tspan = linspace(0,P,n);  
  
    xphi0 = [x0;reshape(eye(6,6),36,1)];  
    [to,xphi] =  
        ode45(@(t,xphi)XSTMEOM(t,xphi,mu),tspan,xphi0,odeset('AbsTol',1e-12,'RelTol',1e-9)  
  
    M = reshape(xphi(end,7:end),6,6);  
  
    [ev,e] = eigs(M); e = diag(e); emag = abs(e);  
    evStab = ev(:,find(emag == min(emag)));  
    evUnstab = ev(:,find(emag == max(emag)));  
  
    if nargin>6  
        if trange(1) ~= 0  
            tspan = [0 trange(1)*P];  
            xphi0 = [x0;reshape(eye(6,6),36,1)];  
            [to,xphi] =  
                ode45(@(t,xphi)XSTMEOM(t,xphi,mu),tspan,xphi0,odeset('AbsTol',1e-12,'RelTol',1e-9)  
                    tspan = linspace(trange(1)*P,trange(2)*P,n);  
                    [to,xphi] =  
                ode45(@(t,xphi)XSTMEOM(t,xphi,mu),tspan,xphi(end,:),odeset('AbsTol',1e-12,'RelTol',1e-9)  
                    else  
                        tspan = linspace(trange(1)*P,trange(2)*P,n);  
                        xphi0 = [x0;reshape(eye(6,6),36,1)];  
                        [to,xphi] =  
                ode45(@(t,xphi)XSTMEOM(t,xphi,mu),tspan,xphi0,odeset('AbsTol',1e-12,'RelTol',1e-9)  
                    end  
        end  
    end  
  
    for i = 1:n  
        x = xphi(i,1:6)';  
        phi = reshape(xphi(i,7:end),6,6);  
  
        x0Sp = x + pert*phi*evStab/norm(phi*evStab);  
        x0USp = x + pert*phi*evUnstab/norm(phi*evUnstab);  
        x0Sm = x - pert*phi*evStab/norm(phi*evStab);  
        x0USm = x - pert*phi*evUnstab/norm(phi*evUnstab);  
  
        tspan = [0 nP*P];%tspan = linspace(0,2*P,1000);  
        [tUSp{i},XUSp{i}] =  
            ode45(@(t,x)JacobiEOM(t,x,mu),tspan,x0USp,odeset('AbsTol',1e-12,'RelTol',1e-9));  
        [tUSm{i},XUSm{i}] =  
            ode45(@(t,x)JacobiEOM(t,x,mu),tspan,x0USm,odeset('AbsTol',1e-12,'RelTol',1e-9));  
        tspan = [0 -nP*P];%tspan = linspace(0,-2*P,1000);  
        [tSp{i},XSp{i}] =  
            ode45(@(t,x)JacobiEOM(t,x,mu),tspan,x0Sp,odeset('AbsTol',1e-12,'RelTol',1e-9));  
        [tSm{i},XSm{i}] =  
            ode45(@(t,x)JacobiEOM(t,x,mu),tspan,x0Sm,odeset('AbsTol',1e-12,'RelTol',1e-9));
```

---

```
end
```

```
end
```

## Poincare Map:

```
function [PD] = Poincare(x,S,SV,CP)
PD = [];
if ~iscell(x)
    points = find(diff(sign(x(:,S)-SV)));
    PD = x(points,CP);
else
    for i = 1:length(x)
        points = find(diff(sign(x{i}(:,S)-SV)));
        PD = [PD; x{i}(points,CP)];
    end
end
end

end
```

## Generate Twin Orbits:

```
function [x0,P] = OrbitalEnergyTrack(Cd,x0L,PL,mu)

C = @(x) 2*((1-mu)./sqrt((x(:,1)+mu).^2+x(:,2).^2+x(:,3).^2)+mu./
sqrt((x(:,1)-1+mu).^2+x(:,2).^2+x(:,3).^2))+x(:,1).^2+x(:,2).^2-
(x(:,4).^2+x(:,5).^2+x(:,6).^2);

tol = 10^-12;
maxiter = 50;
i = 0;

while 1
    xi0 = interp1(C(x0L),x0L(:,1),Cd);
    etadot0 = interp1(C(x0L),x0L(:,5),Cd);
    P = interp1(C(x0L),PL,Cd);
    x0 = [xi0;0;0;0;etadot0;0];

    [XT,P,t] = ShootingMethod(x0,P,mu,[0;0],[1,5],[2,4]);
    x0 = XT{end}(1,1:6);

    x0L(end+1,:) = x0;
    PL(end+1) = P;
    [~,S] = sort(x0L(:,1));
    x0L = x0L(S,:);
    PL = PL(S);

    if abs(C(x0)-Cd)<tol || i>maxiter
        break;
    end
    i = i + 1;
end
```

---

---

```

if i>maxiter
    disp('Energy Tracking Algorithm Did Not Converge Before Maximum
    Iteration Limit');
else
    disp(sprintf('Energy Tracking Algorithm Converged in %d
    iterations',i));
end

end

```

## Generate Transfer Orbits:

```

function [xt,x01,x02,PMUS1,PMS2,P1,P2,P01,P02] =
    TransferOrbit(x0L0,x0Lf,PL0,PLf,Cd,xiP,mu,pert1,nP1,pert2,nP2)

[x01,P01] = OrbitalEnergyTrack(Cd,x0L0,PL0,mu);
[x02,P02] = OrbitalEnergyTrack(Cd,x0Lf,PLf,mu);

trange = [0 1];
n = 1001; % Number Of Manifolds

[XSp1,XUSp1,XSm1,XUSm1,tSp1,tUSp1,tSm1,tUSm1] =
    Manifolds(x01',P01,mu,n,pert1,nP1,trange);
[XSp2,XUSp2,XSm2,XUSm2,tSp2,tUSp2,tSm2,tUSm2] =
    Manifolds(x02',P02,mu,n,pert2,nP2,trange);

XUS1 = [XUSp1,XUSm1];
tUS1 = [tUSp1,tUSm1];
XS2 = [XSp2,XSm2];
tS2 = [tSp2,tSm2];

% Find all manifolds that pass through the Poincare section, save to
cell array
for i = 1:2*n
    P2{i} = XS2{i};
    P1{i} = XUS1{i};
    n2 = find(diff(sign(P2{i}(:,1)-xiP)));
    if ~isempty(n2)
        P2{i}(n2+2:end,:) = [];
        tS2{i}(n2+2:end) = [];
    else
        P2{i} = [];
        tS2{i} = [];
    end
    n1 = find(diff(sign(P1{i}(:,1)-xiP)));
    if ~isempty(n1)
        P1{i}(n1+1:end,:) = [];
        tUS1{i}(n1+1:end) = [];
    else
        P1{i} = [];
        tUS1{i} = [];
    end
end

```

---

```

end

% Find which manifolds produce closest points on Poincare map
min = 10;
minI = 1; minJ = 1;
for i = 1:2*n
    for j = 1:2*n
        if ~isempty(P1{i}) && ~isempty(P2{j}) && norm(P1{i}(end,
[2,5])-P2{j}(end,[2,5]))<min
            minI = i;
            minJ = j;
            min = norm(P1{i}(end,[2,5])-P2{j}(end,[2,5]));
        end
    end
end

% Data for Poincare Map:
PMS2 = []; PMUS1 = [];
for i = 1:2*n
    if ~isempty(P1{i})
        PMUS1(end+1,:) = P1{i}(end,[2,5]);
    end
end
for j = 1:2*n
    if ~isempty(P2{j})
        PMS2(end+1,:) = P2{j}(end,[2,5]);
    end
end

if min>.05
    disp('Transfer Orbit Not Found');
end

xt = [P1{minI};P2{minJ}(end:-1:1,:)];

end

```

*Published with MATLAB® R2020b*