AE 502: Homework Project 3

Sharath Yalla

04/16/2023

04/16 HW 3 pg 1 Yalla Dr. Eggl

Problem 1

Per homework discussion:

Let $\mathcal{H}_D = -\frac{1}{2L^2} + \omega_D H$ (Assumption here that ω in problem is ω_D and not g)

$\mathcal{H}_0 = -\frac{1}{2L^2}$

$\mathcal{H}_1 = \varepsilon \omega_D H$

p	q
$L = \sqrt{\mu a}$	M
$G = L\sqrt{1-e^2}$	ω
$H = G\cos i$	Ω

Slide 4 03/23

- New Hamiltonian?
- Corresponding EOM?

$\{\mathcal{H}_0, S\} + \mathcal{H}_1 = 0$

$\{H_0, S\}$

$\sum_i \frac{\partial H_0}{\partial q_i}\frac{\partial S}{\partial p_i} - \frac{\partial H_0}{\partial p_i}\frac{\partial S}{\partial q_i}$

Assume that $\frac{\partial H_0}{\partial q_i} = 0$

$\Longrightarrow \{H_0, S\} = \sum_i -\frac{\partial H_0}{\partial p_i}\frac{\partial S}{\partial q_i}$

$H_0(L) \Rightarrow$

$\{H_0, S\} = -\frac{\partial H_0}{\partial L}\frac{\partial S}{\partial M}$

$\{H_0, S\} + H_1 =$

$0 = -\frac{\partial H}{\partial L}\frac{\partial S}{\partial M} + \omega_D H$

$0 = L^{-3}\frac{\partial S}{\partial M} + \omega_D H$

$L^{-3}\frac{\partial S}{\partial M} = -\omega_D H$

$\frac{\partial S}{\partial M} = -L^3 \omega_D H$

Ansatz

$S = -L^3 \omega_D H M$

04/16 HW3 pg 2 Yalla Dr. Eggl

$$S = -\omega_D HML^3$$

$$P_i = L_S^{-\varepsilon} p_i = p_i - \varepsilon\{p_i, S\} + \dots$$

$$Q_i = L_S^{-\varepsilon} q_i = q_i - \varepsilon\{q_i, S\} + \dots$$

$\longrightarrow$ This is an assumption on my part

p'

$$L' = L - \varepsilon\{L, S\} + \dots$$

$$G' = G - \varepsilon\{G, S\} + \dots$$

$$H' = H - \varepsilon\{H, S\} + \dots$$

q'

$$M' = M - \varepsilon\{M, S\}$$

$$\omega' = \omega - \varepsilon\{\omega, S\}$$

$$\Omega' = \Omega - \varepsilon\{\Omega, S\}$$

$$\{L, S\} = \cancelto{0}{\frac{\partial L}{\partial M}}\frac{\partial S}{\partial L} - \frac{\partial L}{\partial L}\frac{\partial S}{\partial M}$$

$$= -\frac{\partial S}{\partial M} = \omega_D H L^3$$

$$\{G, S\} = \cancelto{0}{\frac{\partial G}{\partial \omega}}\frac{\partial S}{\partial G} - \frac{\partial G}{\partial G}\cancelto{0}{\frac{\partial S}{\partial \omega}}$$

$$= 0$$

$$\{H, S\} = \cancelto{0}{\frac{\partial H}{\partial \Omega}}\frac{\partial S}{\partial H} - \frac{\partial H}{\partial H}\cancelto{0}{\frac{\partial S}{\partial \Omega}}$$

$$= 0$$

$$\{M, S\} = \frac{\partial M}{\partial M}\frac{\partial S}{\partial L} - \cancelto{0}{\frac{\partial M}{\partial L}}\frac{\partial S}{\partial M}$$

$$= -3\omega_D HML^2$$

$$\{\omega, S\} = \frac{\partial \omega}{\partial \omega}\cancelto{0}{\frac{\partial S}{\partial G}} - \cancelto{0}{\frac{\partial \omega}{\partial G}}\frac{\partial S}{\partial \omega}$$

$$= 0$$

$$\{\Omega, S\} = \frac{\partial \Omega}{\partial \Omega}\frac{\partial S}{\partial H} - \cancelto{0}{\frac{\partial \Omega}{\partial H}}\frac{\partial S}{\Omega}$$

$$= -\omega_D ML^3$$

04/16 HW3 pg 3 Yalla Dr. Eggl

$L' = L - \omega_D H L^3$ $\quad$ $M' = M + 3\omega_D H M L^2$

$G' = G$ $\quad$ $\omega' = \omega$ $\quad$ Old variables

$H' = H$ $\quad$ $\Omega' = \Omega + \omega_D M L^3$ $\quad$ → new variables

$$\boxed{K = \mathcal{H}(L') = \frac{-1}{2L'^2}}$$ New Hamiltonian

---

EOM

$$\dot{q}_i = \frac{\partial H}{\partial p_i}, \quad \dot{p}_i = \frac{-\partial H}{\partial q_i}$$

$$\Rightarrow \dot{Q}_i = \frac{\partial K}{\partial P_i}, \quad \dot{P}_i = \frac{-\partial K}{\partial Q_i}$$

$$\boxed{\begin{aligned} \dot{L}' &= \frac{-\partial K}{\partial M'} = 0 & \dot{M}' &= \frac{\partial K}{\partial L'} = L'^{-3} \\ \dot{G}' &= \frac{-\partial K}{\partial \omega'} = 0 & \dot{\omega}' &= \frac{\partial K}{\partial G'} = 0 \\ \dot{H}' &= \frac{-\partial K}{\partial \Omega'} = 0 & \dot{\Omega}' &= \frac{\partial K}{\partial H'} = 0 \end{aligned}}$$

New E.O.M.

Problem 1: Assume the rotation rate of the frame is very low compared to the velocities in your system, e.g. ω=0.01 per system time unit (TU). Use Hori-Lie-Deprit perturbation theory to push the perturbation beyond the first order in ω. What do the new Kamiltonian and the corresponding equations of motion look like?

Above you can see my utilization of Hori-Lie-Perturbation theory to push the perturbation beyond the first order in ω and generate the new Kamiltonian and corresponding equations of motion. The first step was to set all values of O(ε) to zero. This is done by setting the following:

$$0 = K_1 = \{H_o, S\} + H_1$$

Using that we can determine our generating function S:

$$\frac{\partial \mathrm{S}}{\partial \mathrm{M}} = -\mathrm{L}^3 \omega_{\mathrm{D}} H$$

$$S = -\mathrm{L}^3 \omega_{\mathrm{D}} H M$$

Then we use our known equation to go from the old to new to the old variable. This can be seen below:

$$\mathrm{P_i} = \mathrm{p_i} - \varepsilon\{\mathrm{p_i}, \mathrm{S}\}$$

$$\mathrm{Q_i} = \mathrm{q_i} - \varepsilon\{\mathrm{q_i}, \mathrm{S}\}$$

Utilizing that we get our new transformed variables which are as follows:

$$\mathrm{L}' = L - \omega_{\mathrm{D}} H \mathrm{L}^3$$

$$\mathrm{G}' = G$$

$$\mathrm{H}' = H$$

$$\mathrm{M}' = M + 3\omega_{\mathrm{D}} H M \mathrm{L}^2$$

$$\omega' = \omega$$

$$\Omega' = \Omega + \omega_{\mathrm{D}} M \mathrm{L}^3$$

As the the transformed Kamiltonian is only the 1st order Hamiltonian with the transformed variables, our **Kamiltonian** is the following:

$$K = \mathrm{H_0}(\mathrm{L}') = \frac{-1}{\mathrm{L}'^2}$$

Our Equations of Motion are by definition:

$$\dot{\mathrm{Q}}_\mathrm{i} = \frac{\partial \mathrm{K}}{\partial \mathrm{P_i}}, \dot{\mathrm{P}}_\mathrm{i} = \frac{-\partial \mathrm{K}}{\partial \mathrm{Q_i}}$$

This leads us to the following **E.O.M**:

$$
\begin{aligned}
\dot{L}' &= \frac{-\partial K}{\partial M'} = 0 & \dot{M}' &= \frac{\partial K}{\partial L'} = L'^{-3} \\
\dot{G}' &= \frac{-\partial K}{\partial \omega'} = 0 & \dot{\omega}' &= \frac{\partial K}{\partial G'} = 0 \\
\dot{H}' &= \frac{-\partial K}{\partial \Omega'} = 0 & \dot{\Omega}' &= \frac{\partial K}{\partial H'} = 0
\end{aligned}
$$

04/16 HW 3 pg 4 Yalla Dr. Eggl

Problem 2

$$\mathcal{H}_0 = -\frac{1}{2L^2} + \omega H$$

$L = na^2$ $\quad M$

$G = L(1-e^2)^{1/2}$ $\quad \omega$

$H = G \cos i$ $\quad \Omega$

$\Downarrow$

$a = \sqrt{L/N}$

$e = \sqrt{1-(G/L)^2}$

$i = a\cos(H/G)$

1) Find Delauney over time
2) Convert from Delauney $\longrightarrow$ traditional o.e
3) Convert from traditional oe $\longrightarrow$ $\underline{r}, \underline{v}$

$$\dot{p}_i = -\frac{\partial \mathcal{H}_0}{\partial q_i} \qquad \dot{q}_i = \frac{\partial H_0}{\partial p_i}$$

P

$\dot{L} = -\frac{\partial H}{\partial M} = 0 \Rightarrow L = L_0$ $\qquad \dot{M} = L^{-3} = \frac{\partial H}{\partial L}$

$\dot{G} = -\frac{\partial H}{\partial \omega} = 0$ $\qquad \dot{\omega} = \frac{\partial H}{\partial G} = 0$

$\dot{H} = \frac{-\partial H}{\partial \Omega} = 0$ $\qquad \dot{\Omega} = \frac{\partial \mathcal{H}}{\partial H} = \omega_D$

$\Downarrow$

$L = L_0$ $\qquad M = L^{-3} t + M_0$

$G = G_0$ $\qquad \omega = \omega_0$

$H = H_0$ $\qquad \Omega = \omega_D t + \Omega_0$

Problem 2: A more convenient representation of the same system can be achieved through the use of Delaunay variables. In fact, the above Hamiltonian expressed in Delaunay variables simply reads: ...

Determine the equations of motion and plot the orbit of the satellite in Cartesian, non-rotating space, i.e. r(t), for a=1, e=0.5, i=45° for 100 time units.

As with Problem the EOM can be simply determined by utilizing the following equations:

$$\dot{Q}_i = \frac{\partial K}{\partial P_i}\ , \dot{P}_i = \frac{-\partial K}{\partial Q_i}$$

This leads us to the following:

$$\begin{aligned}
\dot{L} &= -\frac{\partial H}{\partial M} = 0 \Rightarrow L = L_0 & \dot{M} &= L^{-3} = \frac{\partial H}{\partial L} \\
\dot{G} &= -\frac{\partial H}{\partial \omega} = 0 & \dot{\omega} &= \frac{\partial H}{\partial G} = 0 \\
\dot{H} &= \frac{-\partial H}{\partial \Omega} = 0 & \dot{\Omega} &= \frac{\partial H}{\partial H} = \omega_D
\end{aligned}$$

Utilizing our equations of motions we can see the evolution of Delauney variables over time, convert them into our “traditional” orbital elements and then convert them into the position vector in Cartesian coordinates. The result of this can be seen below and the code to plot such will be attached to this document and uploaded to the Git repository.

```
clear
```

Givens from problem

```
a = 1; e = .5; i = 45*pi/180; t = [0:.001:100]; mu = 1;
```

Assuming that M_0, w_0, and O_0 are equal to 0 and that w_d = .01

```
L = sqrt(a); G = L*sqrt(1 - e^2); H = G*cos(i);
M_0 = 0; w_0 = 0; O_0 = 0; w_d = .01;
M = L^(-3).*t + M_0; w = w_0; O = w_d.*t + O_0;

n = sqrt(mu/(a^3));
```

Go from Delauney to traditional OEs, some of the below calculations wil be trivial and redundant

```
a_t = sqrt(L./n);
e_t = sqrt(1 - (G ./ L).^2);
i_t = acos(H./G);
M_t = M;
w_t = w;
O_t = O;
```

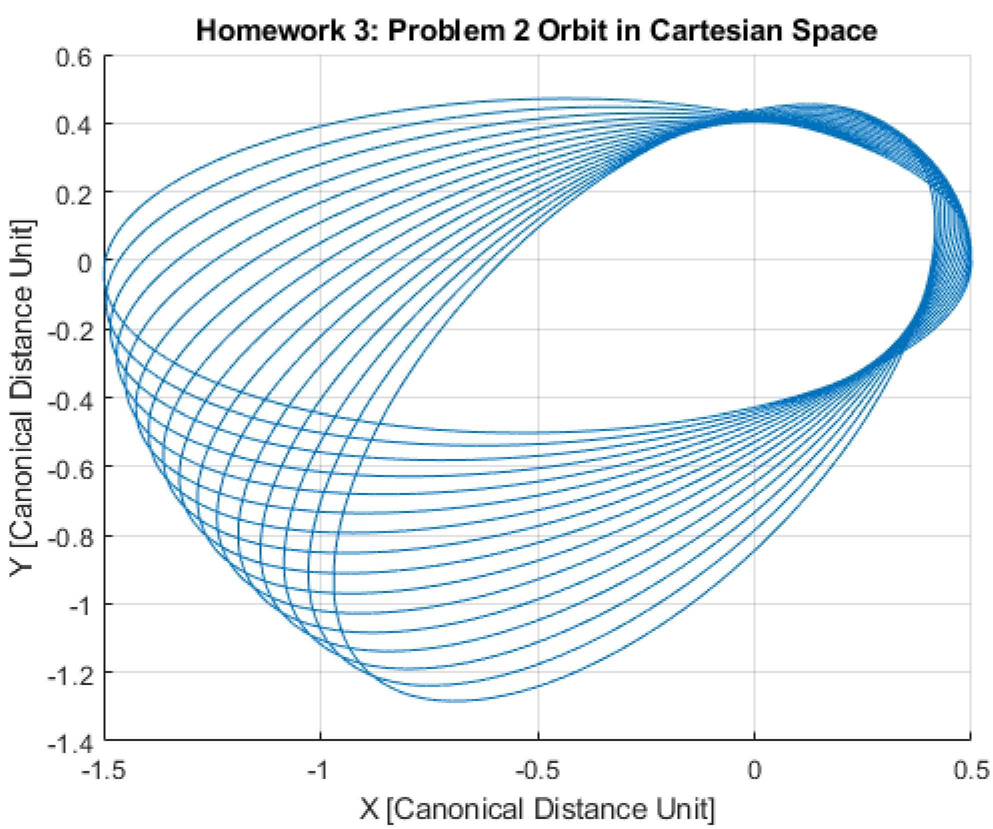
Converting from traditional OEs to r_ and v_

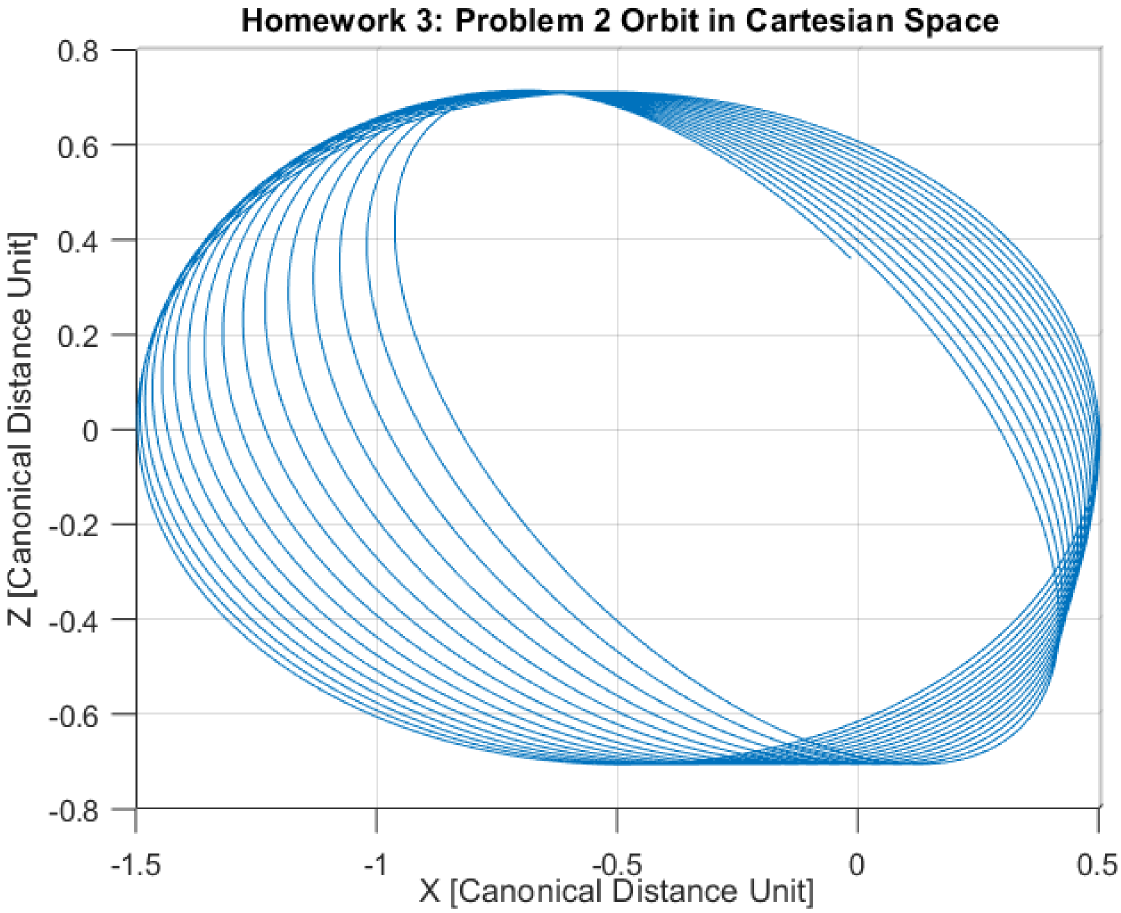
Convert Mean Anomaly to True Anamoly

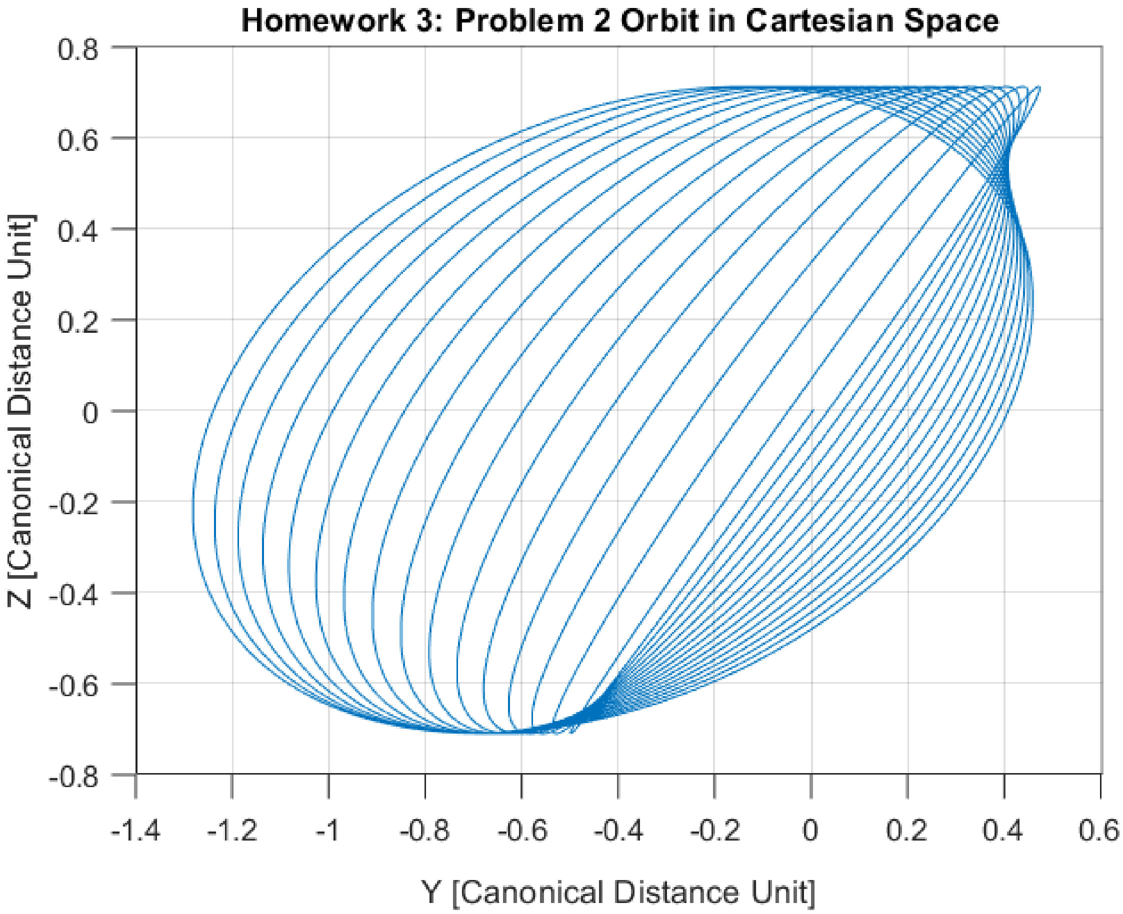
Implementation of Laguerre-Conway: My own code from a previous class

```
iter = 5;
E = M_t;

del = 1000000000;
count = 0;
for(i = 1:length(M_t))
    while(del > .000001)
        F = E(i) - e*sin(E(i)) - M(i);
        F_1 = 1 - e*cos(E(i));
        F_2 = e*sin(E(i));
```

```
        E_old = E(i);
        %Laguerre-Conway
        E(i) = E(i) - (n * F)/ ...
            (F_1+sign(F_1)*abs((n-1)^2*(F_1)^2 - n*(n-1)*F*F_2)^.5);
        del = (E(i) - E_old);
        count  = count + 1;
    end
end
% True Anamoly from eccentric anamoly
temp = tan(E./2) .* sqrt( (1 + e) / (1 - e) );
TA = 2 .* atan(temp);
mag_r = a*(1 - e.*cos(E));
h = sqrt(mu*a*(1 - e^2));
r_X = mag_r.*(cos(O_t).*cos(w_t + TA) - sin(O_t).*sin(w_t + TA)*cos(i));
r_Y = mag_r.*(sin(O_t).*cos(w_t + TA) + cos(O_t).*sin(w_t + TA)*cos(i));
r_Z = mag_r.*(sin(i) .* sin(w_t + TA));

figure
hold on
plot3(r_X, r_Y, r_Z)
xlabel('X [Canonical Distance Unit]'); ylabel('Y [Canonical Distance Unit]');
zlabel('Z [Canonical Distance Unit]'); grid on; title('Homework 3: Problem 2 Orbit in Cartesian
Space')
hold off
```




Homework 3: Problem 2 Orbit in Cartesian Space
Z [Canonical Distance Unit]
X [Canonical Distance Unit]
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1.5
-1
-0.5
0
0.5



Published with MATLAB® R2022a

Problem 3 (Bonus): (BONUS QUESTION - no partial credit - 50 points) Compare analytic solutions of question 1 to those of question 2 for at least 20 different initial conditions and integration times of 100 time units (TU) in two equinoctial element plots, namely h vs k and p vs q, where

$h = e \cdot \sin(\omega + \Omega)$ (8)

$k = e \cdot \cos(\omega + \Omega)$ (9)

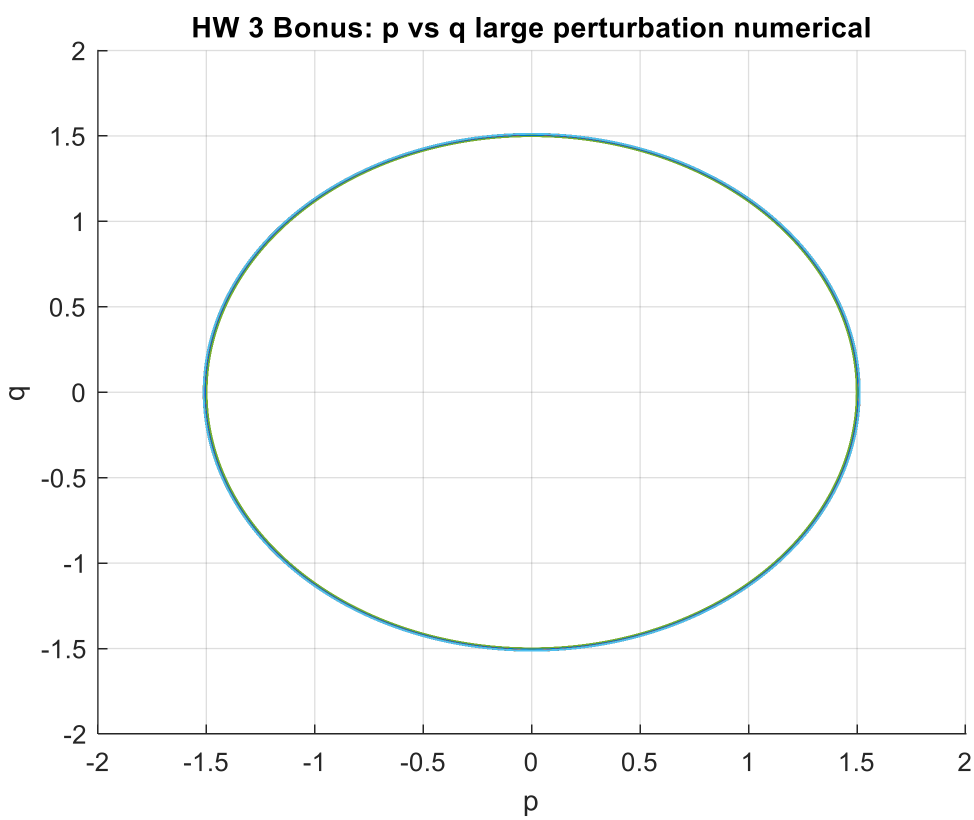
$p = \tan(i/2) \sin \Omega$ (10)

$q = \tan(i/2) \cos \Omega$ (11)

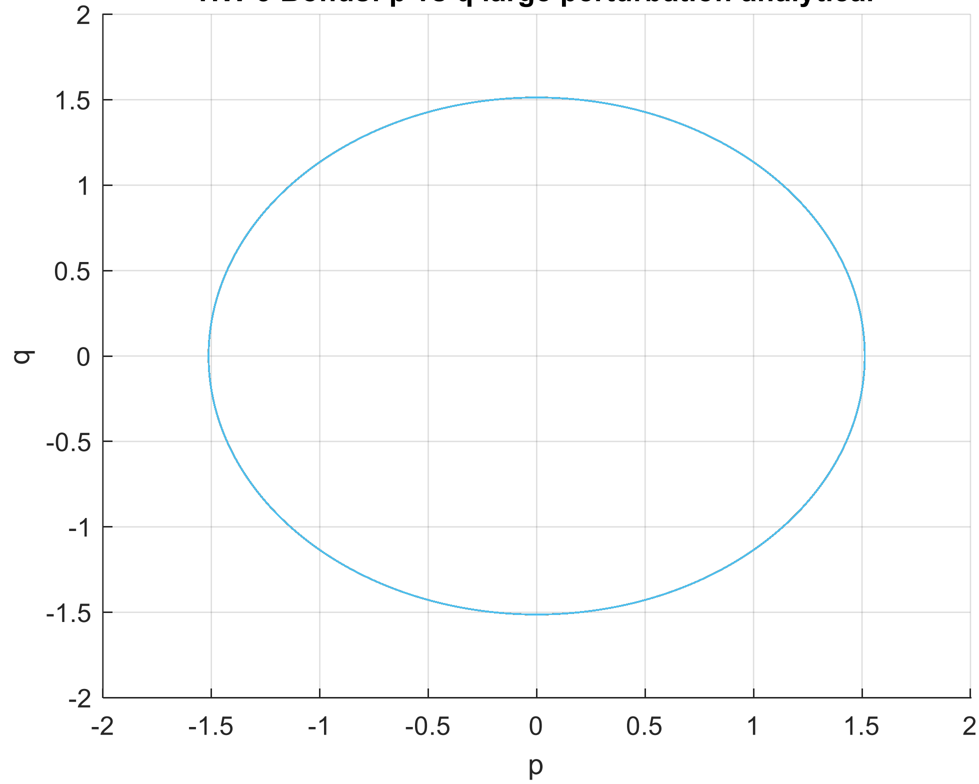
Reproduce the same plots for frame rotation frequencies $\omega = \{0.02, 0.1, 0.5\}$ TU^{-1}

What can you say about the behavior of the analytic and perturbation solutions?

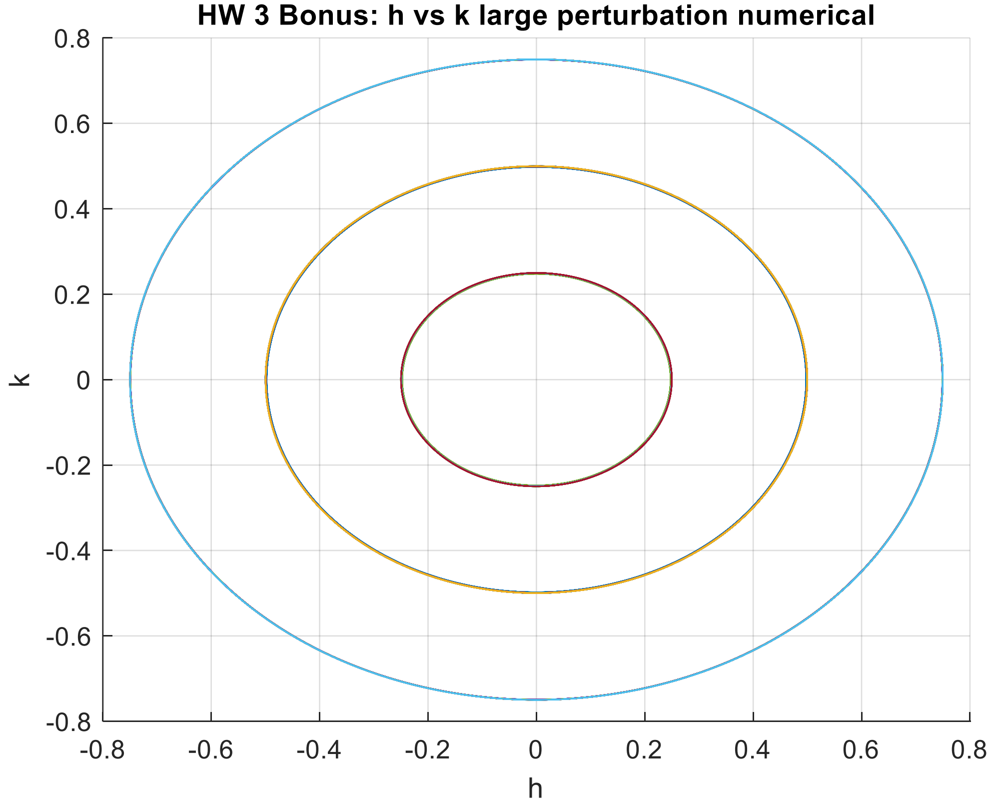
As one can see below, the plots look fairly similar between the analytic and perturbation solutions especially for small perturbations. This frankly seems to not make sense as the analytic solution should be poorly behaved for the large perturbation. The plots however are presented below:

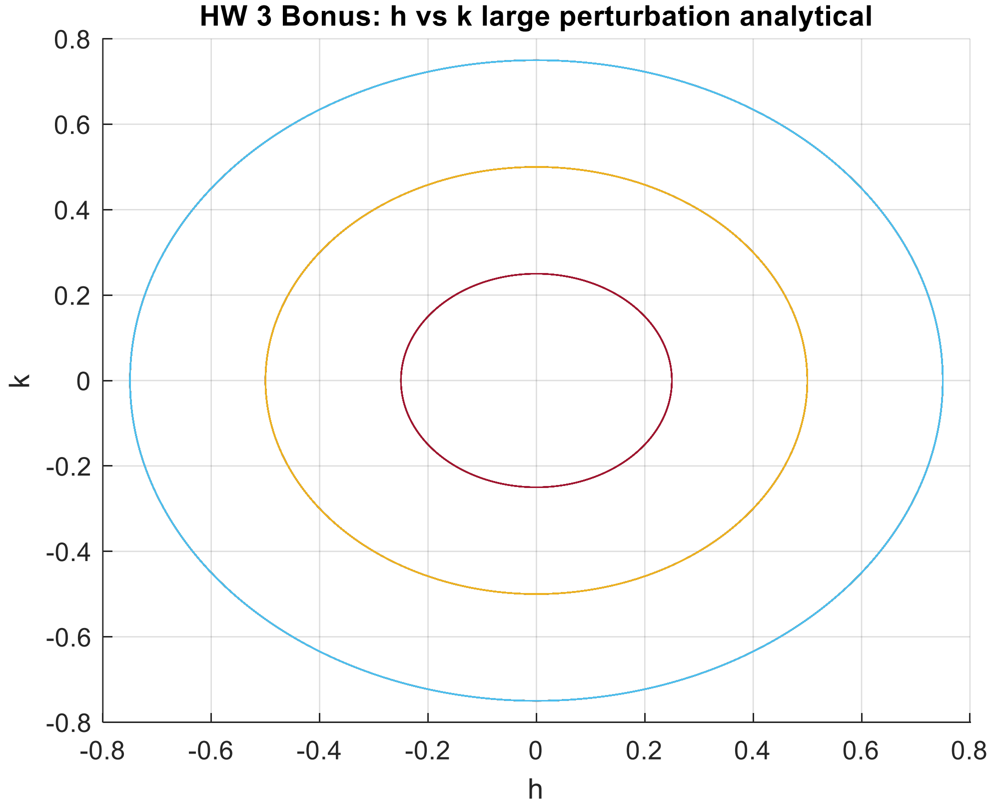


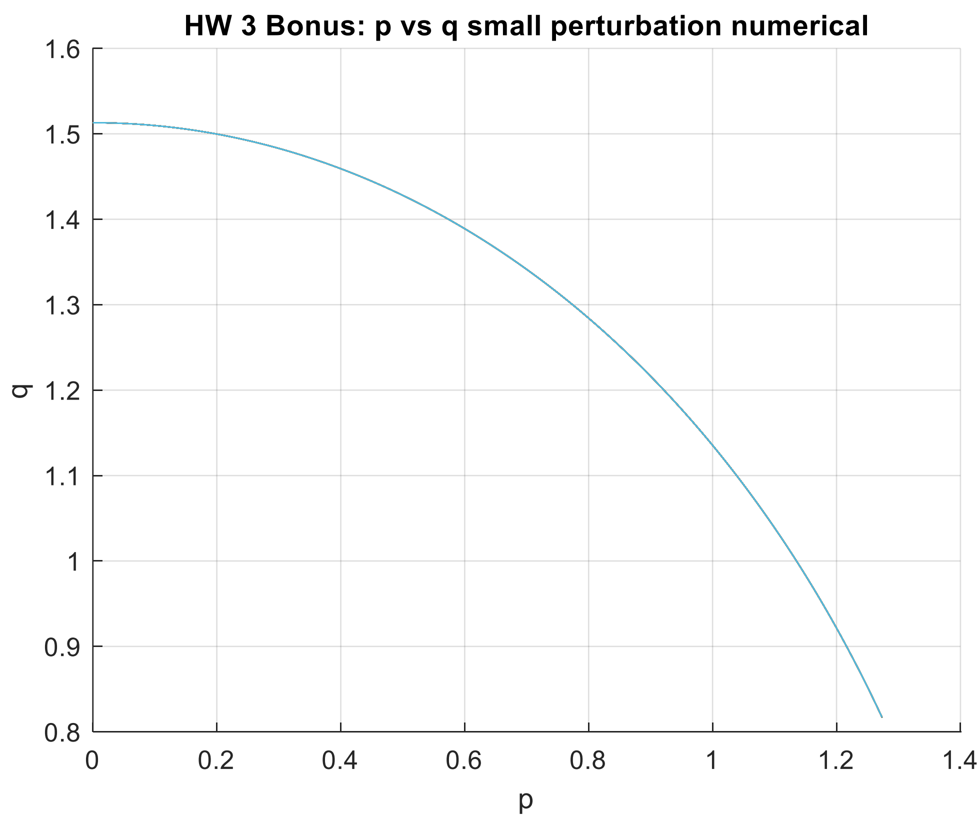
HW 3 Bonus: p vs q large perturbation analytical

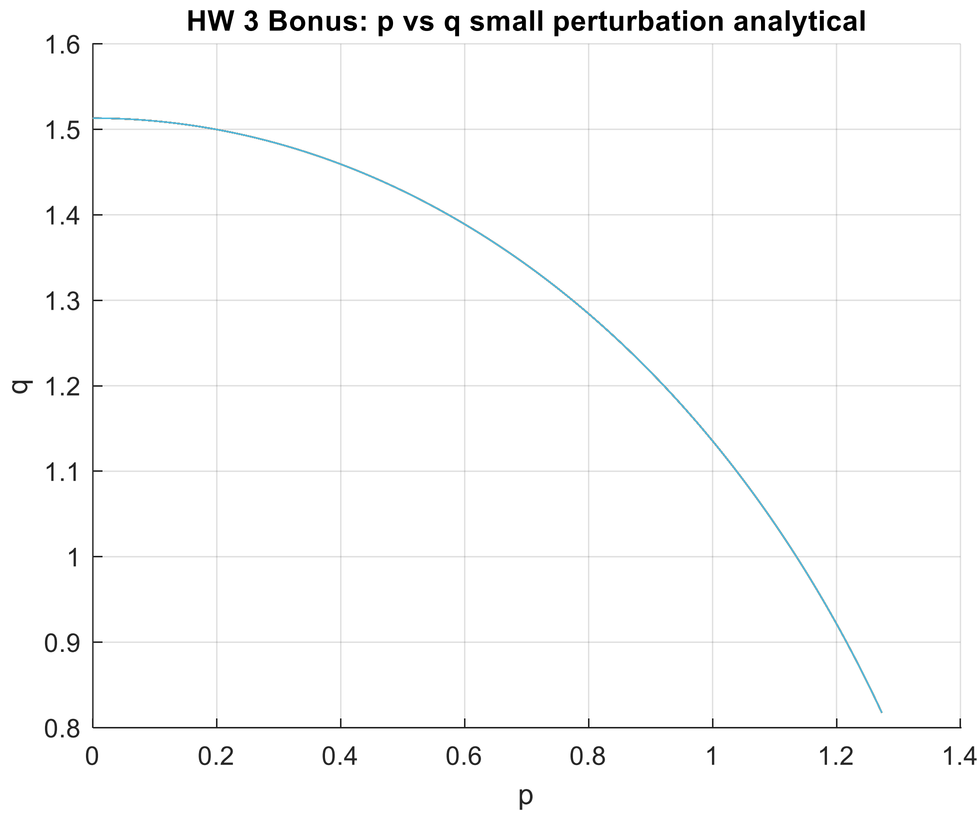


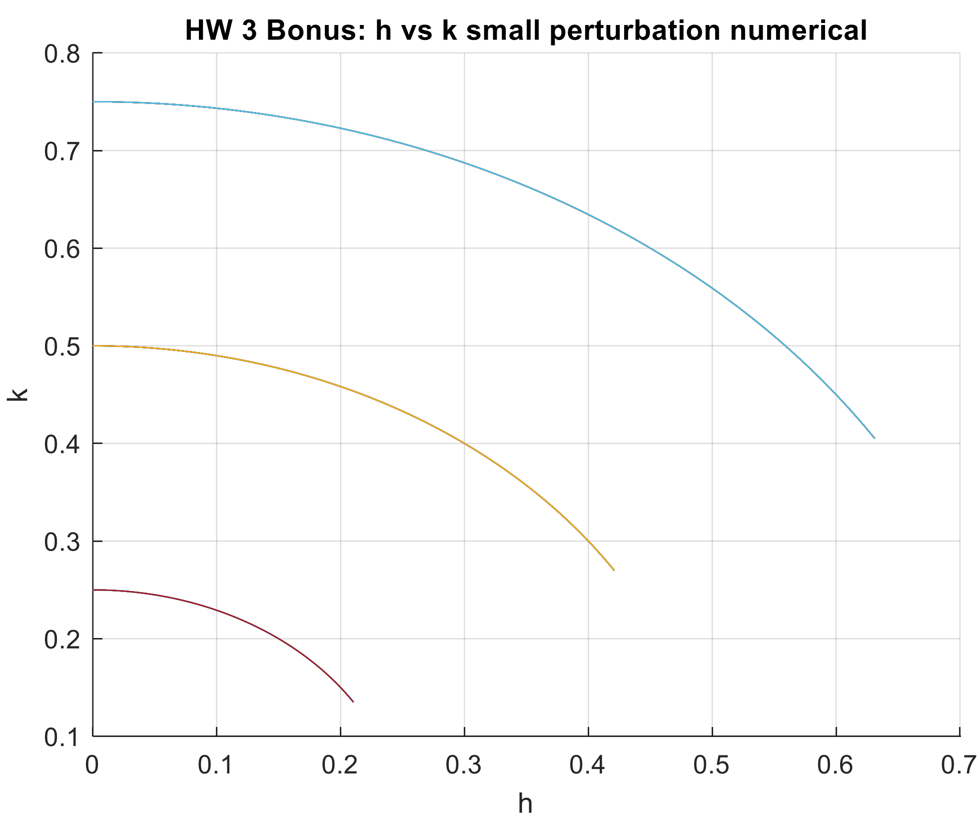
HW 3 Bonus: h vs k large perturbation numerical

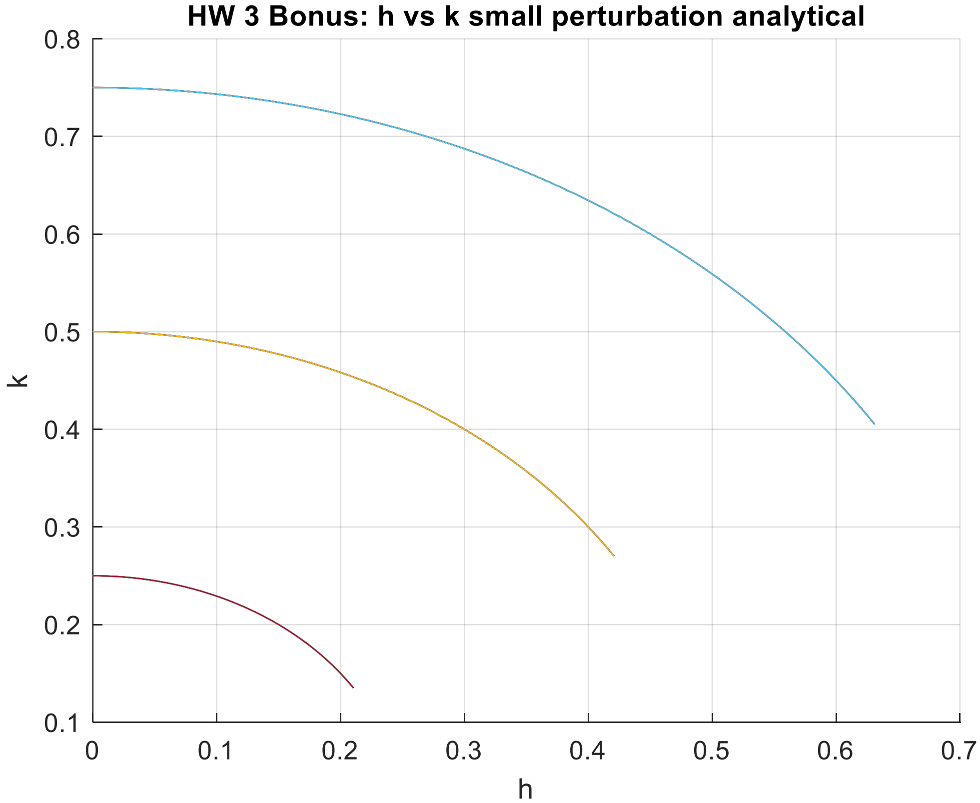



HW 3 Bonus: h vs k large perturbation analytical
k
h
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8


HW 3 Bonus: p vs q small perturbation numerical
q
p
1.6
1.5
1.4
1.3
1.2
1.1
1
0.9
0.8
0
0.2
0.4
0.6
0.8
1
1.2
1.4


HW 3 Bonus: p vs q small perturbation analytical
q
p
0
0.2
0.4
0.6
0.8
1
1.2
1.4
0.8
0.9
1
1.1
1.2
1.3
1.4
1.5
1.6


HW 3 Bonus: h vs k small perturbation numerical
k
h
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8


HW 3 Bonus: h vs k small perturbation analytical
k
h
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7

Code:

Github Link: https://github.com/syalla2/AE502_HW3

Raw Code:

hw3_2.m:

```
clear
%% Givens from problem
a = 1; e = .5; i = 45*pi/180; t = [0:.001:100]; mu = 1;

%% Assuming that M_0, w_0, and O_0 are equal to 0 and that w_d = .01
L = sqrt(a); G = L*sqrt(1 - e^2); H = G*cos(i);
M_0 = 0; w_0 = 0; O_0 = 0; w_d = .01;
M = L^(-3).*t + M_0; w = w_0; O = w_d.*t + O_0;

n = sqrt(mu/(a^3));

%% Go from Delauney to traditional OEs, some of the below calculations will
%  be trivial and redundant
a_t = sqrt(L./n);
e_t = sqrt(1 - (G ./ L).^2);
i_t = acos(H./G);
M_t = M;
w_t = w;
O_t = O;

%% Converting from traditional OEs to r_ and v_
%% Convert Mean Anomaly to True Anamoly
%% Implementation of Laguerre-Conway: My own code from a previous class
iter = 5;
E = M_t;

del = 1000000000;
count = 0;
for(i = 1:length(M_t))
    while(del > .000001)
        F = E(i) - e*sin(E(i)) - M(i);
        F_1 = 1 - e*cos(E(i));
        F_2 = e*sin(E(i));
        E_old = E(i);
        %Laguerre-Conway
        E(i) = E(i) - (n * F)/ ...
            (F_1+sign(F_1)*abs((n-1)^2*(F_1)^2 - n*(n-1)*F*F_2)^.5);
        del = (E(i) - E_old);
        count  = count + 1;
    end
end
% True Anamoly from eccentric anamoly
temp = tan(E./2) .* sqrt( (1 + e) / (1 - e) );
TA = 2 .* atan(temp);
mag_r = a*(1 - e.*cos(E));
```

```
h = sqrt(mu*a*(1 - e^2));
r_X = mag_r.*(cos(O_t).*cos(w_t + TA) - sin(O_t).*sin(w_t + TA)*cos(i));
r_Y = mag_r.*(sin(O_t).*cos(w_t + TA) + cos(O_t).*sin(w_t + TA)*cos(i));
r_Z = mag_r.*(sin(i) .* sin(w_t + TA));

figure
hold on
plot3(r_X, r_Y, r_Z)
xlabel('X [Canonical Distance Unit]'); ylabel('Y [Canonical Distance Unit]');
zlabel('Z [Canonical Distance Unit]'); grid on; title('Homework 3: Problem 2 Orbit in
Cartesian Space')
hold off
```

hw3_bonus.m:

```
clear
a_ = [.75 1 1.25];
e_ = [.25 .5 .75];
i_ = [30  45  60]*pi/180;
w_d_new = [.02 .1 .5];
w_d_ = [.01 norm(w_d_new)];
t = [0:.1:100];
mu = 1;

e_final1 =  [];
O_final1 =  [];
i_final1  = [];

e_final2 =  [];
O_final2 =  [];
i_final2  = [];

for i_w = 1:1:2
    for i_a = 1:1:3
        for i_e = 1:1:3
            for i_i = 1:1:3
                a = a_(i_a); e = e_(i_e); i = i_(i_i); w_d = w_d_(i_w);

                %% Assuming that M_0, w_0, and O_0 are equal to 0 and that w_d = .01
                L = sqrt(a); G = L*sqrt(1 - e^2); H = G*cos(i);
                M_0 = 0; w_0 = 0; O_0 = 0;
                M = L^(-3).*t + M_0; w = w_0; O = w_d.*t + O_0;

                n = sqrt(mu/(a^3));

                %% Go from Delauney to traditional OEs, some of the below
calculations will
                %  be trivial and redundant
                a_t = sqrt(L./n);
                e_t = sqrt(1 - (G ./ L).^2);
                i_t = acos(H./G);
                M_t = M;
                w_t = w;
```

```
                O_t = O;

                %% Converting from traditional OEs to r_ and v_
                %% Convert Mean Anomaly to True Anamoly
                %% Implementation of Laguerre-Conway: My own code from a previous
class
                iter = 5;
                E = M_t;

                del = 1000000000;
                count = 0;
                for(i = 1:length(M_t))
                    while(del > .000001)
                        F = E(i) - e*sin(E(i)) - M(i);
                        F_1 = 1 - e*cos(E(i));
                        F_2 = e*sin(E(i));
                        E_old = E(i);
                        %Laguerre-Conway
                        E(i) = E(i) - (n * F)/ ...
                            (F_1+sign(F_1)*abs((n-1)^2*(F_1)^2 - n*(n-1)*F*F_2)^.5);
                        del = (E(i) - E_old);
                        count  = count + 1;
                    end
                end
                % True Anamoly from eccentric anamoly
                temp = tan(E./2) .* sqrt( (1 + e) / (1 - e) );
                TA = 2 .* atan(temp);

                e_final1 =  [e_final1; e];
                O_final1 =  [O_final1; O_t];
                i_final1  = [i_final1; i];


            end
        end
    end
end


for i_w = 1:1:2
    for i_a = 1:1:3
        for i_e = 1:1:3
            for i_i = 1:1:3
                a = a_(i_a); e = e_(i_e); i = i_(i_i); w_d = w_d_(i_w);

                n = sqrt(mu/(a^3));
                %% My assumptions
                w_0 = 0; M1_0 = 0; O1_0 = 0;

                %% Assuming that M_0, w_0, and O_0 are equal to 0 and that w_d = .01
                L = sqrt(a); G = L*sqrt(1 - e^2); H = G*cos(i);

                L1 = L - w_d*H*L^3;
                G1 = G;
                H1 = H;
```

```
                w1 = w_0;
                M1_t = [M1_0];
                O1_t = [O1_0];

                dt = .1;


                for (i = 2:1:length(t))
                   M_new = M1_t(i - 1) + L1^(-3)*dt;
                   M1_t = [M1_t; M_new];
                end

                M_t = M1_t./( 1 + 3*w_d*H*L^2);
                O_t = -w_d.*M_t.*L^3;

                %% Go from Delauney to traditional OEs, some of the below
calculations will
                %  be trivial and redundant
                a_t = sqrt(L./n);
                e_t = sqrt(1 - (G ./ L).^2);
                i_t = acos(H./G);
                M_t;
                w_t = w_0;
                O_t;

                e_final2 =  [e_final2; e];
                O_final2 =  [O_final2; O_t'];
                i_final2  = [i_final2; i];

            end
        end
    end
end
%% small perturbations
figure
hold on
title('HW 3 Bonus: h vs k small perturbation analytical')
for iter = 1:1:27
    h1 = e_final1(iter).*sin(O_final1(iter, :));
    k1 = e_final1(iter).*cos(O_final1(iter, :));
    plot(h1, k1);
end
xlabel(['h']); ylabel(['k']); grid on
hold off

figure
hold on
title('HW 3 Bonus: h vs k small perturbation numerical')
for iter = 1:1:27
    % Adding a -1 to h2, math error present somewhere
    h2 = e_final2(iter).*sin(O_final2(iter, :))*-1;
    k2 = e_final2(iter).*cos(O_final2(iter, :));
    plot(h2, k2);
end
xlabel(['h']); ylabel(['k']); grid on
```

```
hold off

figure
hold on
title('HW 3 Bonus: p vs q small perturbation analytical')
for iter = 1:1:27
    p1 = tan(i_final1(iter) / 2) .* sin(O_final1(iter, :));
    q1 = tan(i_final1(iter) / 2) .* cos(O_final1(iter, :));
    plot(p1, q1);
end
xlabel(['p']); ylabel(['q']); grid on
hold off

figure
hold on
title('HW 3 Bonus: p vs q small perturbation numerical')
for iter = 1:1:27
    % Adding a -1 to p2, math error present somewhere
    p2 = tan(i_final2(iter) / 2) .* sin(O_final2(iter, :)).*-1;
    q2 = tan(i_final2(iter) / 2) .* cos(O_final2(iter, :));
    plot(p2, q2);
end
xlabel(['p']); ylabel(['q']); grid on
hold off

%% large perturbations

figure
hold on
title('HW 3 Bonus: h vs k large perturbation analytical')
for iter = 28:1:54
    h1 = e_final1(iter).*sin(O_final1(iter, :));
    k1 = e_final1(iter).*cos(O_final1(iter, :));
    plot(h1, k1);
end
xlabel(['h']); ylabel(['k']); grid on
hold off

figure
hold on
title('HW 3 Bonus: h vs k large perturbation numerical')
for iter = 28:1:54
    % Adding a -1 to h2, math error present somewhere
    h2 = e_final2(iter).*sin(O_final2(iter, :))*-1;
    k2 = e_final2(iter).*cos(O_final2(iter, :));
    plot(h2, k2);
end
xlabel(['h']); ylabel(['k']); grid on
hold off

figure
hold on
title('HW 3 Bonus: p vs q large perturbation analytical')
for iter = 28:1:54
    p1 = tan(i_final1(iter) / 2) .* sin(O_final1(iter, :));
```

```
    q1 = tan(i_final1(iter) / 2) .* cos(O_final1(iter, :));
    plot(p1, q1);
end
xlabel(['p']); ylabel(['q']); grid on
hold off

figure
hold on
title('HW 3 Bonus: p vs q large perturbation numerical')
for iter = 28:1:54
    % Adding a -1 to p2, math error present somewhere
    p2 = tan(i_final2(iter) / 2) .* sin(O_final2(iter, :)).*-1;
    q2 = tan(i_final2(iter) / 2) .* cos(O_final2(iter, :));
    plot(p2, q2);
end
xlabel(['p']); ylabel(['q']); grid on
hold off
```