```
%Greg Soos
%Homework 4
%AERO 215-03
%11.07.2017

close all; clear all; clc

mu = 398600 ; %km^3/s^2
R = [ -5096.8 , 3185.5 , 3185.5 ] ; %km
V = [ -3.95 , -6.32 , 4.74 ] ; %km/s

%Classical Orbital Elements - Original Orbit
[ a, e, inc, RAAN, argper, truanom ] = CoEs(R, V) ;
t = 2*pi*sqrt(a^3/mu) ;
disp('--------------')
disp('ORIGINAL ORBIT')
disp('--------------')
disp(['Semi-Major Axis: ', num2str(a), ' km'])
disp(['Eccentricity: ', num2str(e)])
disp(['Inclination: ', num2str(inc), ' degrees'])
disp(['Right Ascension of Ascending Node: ', num2str(RAAN), '
 degrees'])
disp(['Argument of Periapse: ', num2str(argper), ' degrees'])
disp(['True Anomaly: ', num2str(truanom), ' degrees'])
disp(['Period: ', num2str(t), ' seconds'])

%Use CoEs function for orbital elements, calculate period, display
 values.


%Method 1 - Circularize Orbit @ Apogee, Simple Plane Change, HT to GEO
Ra = a*(1+e) ; %km
V_ap = sqrt(mu*(2/Ra - 1/a)) ;
t = 2*pi*sqrt(Ra^3/mu) ;
DeltaV1 = abs(sqrt(mu*(2/Ra - 1/a)) - (2*pi*Ra/t)) ; %Burn 1
V_b1 = V_ap + DeltaV1 ;
disp('--------------')
disp('Method 1: ')
disp(['Delta V of Burn 1: ', num2str(DeltaV1), ' km/s'])
disp(['Velocity After Burn 1: ', num2str(V_b1), ' km/s'])

%Calculate radius of apogee, velocity at apogee, period of orbit, and
 use
%to find change in velocity. Use change in velocity to find final
 velocity.
%Display values.

V2 = sqrt(mu/Ra) ;
DeltaV2 = abs(sqrt(2*V2^2 - 2*V2^2*cosd(inc))) ; %Burn 2
disp(['Delta V of Burn 2: ', num2str(DeltaV2), ' km/s'])
disp(['Velocity After Burn 2: ', num2str(V_b1), ' km/s'])
```

```
%Calculate velocity at new circular orbit, use to calculate change in
%velocity for simple plane change, display values (velocity is the
 same
%because even though the vector changed, it will still be the same
%magnitude since it is still the same size circular orbit).

r_final = 42157 ; %km
a_transfer = (Ra + r_final)/2 ;
DeltaV3 = abs(sqrt(mu*(2/Ra - 1/a_transfer)) - (2*pi*Ra/t)) ; %Burn 3
V_b3 = sqrt(mu*(2/r_final - 1/a_transfer)) ;
disp(['Delta V of Burn 3: ', num2str(DeltaV3), ' km/s'])
disp(['Velocity Before Burn 4: ', num2str(V_b3), ' km/s'])

%Calculate semi-major axis of transfer orbit, use to calculate change
 in
%velocity. Use change in velocity to find final velocity. Display
 values.

t = 2*pi*sqrt(r_final^3/mu) ;
DeltaV4 = abs(sqrt(mu*(2/r_final - 1/a_transfer)) - (2*pi*r_final/
t)) ; %Burn 4
V_b4 = V_b3 + DeltaV4 ;
V_HTtot = DeltaV3 + DeltaV4 ;
disp(['Delta V of Burn 4: ', num2str(DeltaV4), ' km/s'])
disp(['Total Delta V of Hohmann Transfer: ', num2str(V_HTtot), ' km/
s'])
disp(['Velocity After Burn 4: ', num2str(V_b4), ' km/s'])

%Calculate period of final orbit, use to calculate change in velocity.
 Use
%change in velocity to find final velocity. Display values.

%Method 2 - Combined Plane Change (Circularize) @ Perigee, HT to GEO
Rp = a*(1-e) ; %km
V_pe = sqrt(mu*(2/Rp - 1/a)) ;
V_final = sqrt(mu*(2/Rp - 1/Rp)) ;
DeltaV1 = abs(sqrt(V_pe^2 + V_final^2 -
 2*V_pe*V_final*cosd(inc))) ; %Burn 1
disp('--------------')
disp('Method 2: ')
disp(['Delta V of Burn 1: ', num2str(DeltaV1), ' km/s'])
disp(['Velocity After Burn 1: ', num2str(V_final), ' km/s'])

%Calculate radius of perigee, velocity at perigee, and velocity at
 final
%orbit. Use to find change in velocity. Display values.

t = 2*pi*sqrt(Rp^3/mu) ;
a_transfer = (Rp + r_final)/2 ;
DeltaV2 = abs(sqrt(mu*(2/Rp - 1/a_transfer)) - (2*pi*Rp/t)) ; %Burn 2
V_b2 = sqrt(mu*(2/Rp - 1/a_transfer)) ;
disp(['Delta V of Burn 2: ', num2str(DeltaV2), ' km/s'])
disp(['Velocity After Burn 2: ', num2str(V_b2), ' km/s'])
```

```
%Calculate period of initial orbit and semi-major axis of transfer
 orbit.
%Use to calculate change in velocity and final velocity. Display
 values.

t = 2*pi*sqrt(r_final^3/mu) ;
DeltaV3 = abs(sqrt(mu*(2/r_final - 1/a_transfer)) - (2*pi*r_final/
t)) ; %Burn 3
V_b3 = sqrt(mu*(2/r_final - 1/r_final)) ;
V_HTtot = DeltaV2 + DeltaV3 ;
disp(['Delta V of Burn 3: ', num2str(DeltaV3), ' km/s'])
disp(['Total Delta V of Hohmann Transfer: ', num2str(V_HTtot), ' km/
s'])
disp(['Velocity After Burn 3: ', num2str(V_b3), ' km/s'])

%Calculate period of final orbit. Use to calculate change in velocity
 and
%final velocity. Display values.

%Classical Orbital Elements - Final Orbit
R = [ 42157, 0, 0 ] ; %km
V = [ 0 , 3.07, 0 ] ; %km/s
[ a, e, inc, RAAN, argper, truanom ] = CoEs(R, V) ;
t = 2*pi*sqrt(a^3/mu) ;
disp('---------------')
disp('  FINAL ORBIT')
disp('---------------')
disp(['Semi-Major Axis: ', num2str(a), ' km'])
disp(['Eccentricity: ', num2str(e)])
disp(['Inclination: ', num2str(inc), ' degrees'])
disp(['Right Ascension of Ascending Node: ', num2str(RAAN), '
 degrees'])
disp(['Argument of Periapse: ', num2str(argper), ' degrees'])
%%%NOTE: Both the RAAN and argument of perigee are displayed as "NaN"
%%%because both of their calculations divide zero by zero. For RAAN,
 it
%%%divides the x-component of the n-vector by its magnitude, which
 both are
%%%zero since the n-vector is the cross product of angular momentum
 and
%%%K-direction vectors (which are both parallel since this is an ideal
%%%case: zero inclination, circular orbit). Likewise, the argument of
%%%perigee includes the eccentricity in the numerator and denominator
 of
%%%its equation, which--since this is circular so eccentricity is
%%%zero--also gives NaN.
disp(['True Anomaly: ', num2str(truanom), ' degrees'])
disp(['Period: ', num2str(t), ' seconds'])

%Establish new position and velocity vectors. The magnitudes of these
%vectors are given and we also know that the inclination of this orbit
 is
%zero degrees, hence we can logically establish that the spacecraft
 will
```

```
%stay in the xy-plane and thus pass through the x-axis at some point.
 We
%also know that the velocity vector will be normal to the position
 vector,
%and also confined to the xy-plane. Hence, we choose the above values.
 We
%then use these to calculate the period and classic orbit elements of
 the
%final orbit, and display the values.
```

```
--------------
ORIGINAL ORBIT
--------------
Semi-Major Axis: 10173.0366 km
Eccentricity: 0.40744
Inclination: 39.6208 degrees
Right Ascension of Ascending Node: 108.189 degrees
Argument of Periapse: 354.6129 degrees
True Anomaly: 52.6387 degrees
Period: 10211.4434 seconds
--------------
Method 1:
Delta V of Burn 1: 1.2147 km/s
Velocity After Burn 1: 5.2763 km/s
Delta V of Burn 2: 3.5764 km/s
Velocity After Burn 2: 5.2763 km/s
Delta V of Burn 3: 1.1706 km/s
Velocity Before Burn 4: 2.1896 km/s
Delta V of Burn 4: 0.88534 km/s
Total Delta V of Hohmann Transfer: 2.0559 km/s
Velocity After Burn 4: 3.0749 km/s
--------------
Method 2:
Delta V of Burn 1: 6.1917 km/s
Velocity After Burn 1: 8.1316 km/s
Delta V of Burn 2: 2.6249 km/s
Velocity After Burn 2: 10.7565 km/s
Delta V of Burn 3: 1.5368 km/s
Total Delta V of Hohmann Transfer: 4.1617 km/s
Velocity After Burn 3: 3.0749 km/s
---------------
  FINAL ORBIT
---------------
Semi-Major Axis: 42022.6361 km
Eccentricity: 0.0031974
Inclination: 0 degrees
Right Ascension of Ascending Node: NaN degrees
Argument of Periapse: NaN degrees
True Anomaly: 180 degrees
Period: 85730.6585 seconds
```

Published with MATLAB® R2017a