```
1 import numpy as np
 2 from scipy import integrate
 3 from mpl toolkits.mplot3d import Axes3D
 4 from HW1 import convert rv kep, kepler J2 ODE
 5 import matplotlib.pyplot as plt
7 # constants
 8 params = \{'mu \ E': 398600.0,
             'J2': 1.082626925638815e-03,
10
             'R E': 6378.1363,
11
             'n objs': 2}
12
13 # orbit elems of first satellite
14 a = 10000.0 # [km]
15 e = 0.001 \# [km]
16 inc = 40.0 # [deg]
17 Omega = 80.0 # [deg]
18 omega = 40.0 # [deg]
19 M \emptyset = 0.0 \# [deg]
20 elems_1 = np.array([a, e, inc, Omega, omega, M_0])
21 # convert to state vector
22 x 1 = convert rv kep.convert rv kep('keplerian', elems 1, 0
   .0, 'state') reshape(6)
23
24 # orbit elems of second satellite
25 a = 10000.0 \# [km]
26 e = 0.8 \# [km]
27 inc = 90.0 # [deg]
28 Omega = 80.0 # [deg]
29 omega = 40.0 # [deg]
30 \text{ M}\_0 = 0.0 \# [deg]
31 elems_2 = np.array([a, e, inc, Omega, omega, M_0])
32 # convert to state vector
33 x_2 = convert_rv_kep.convert_rv_kep('keplerian', elems_2, 0
   .0, 'state') reshape(6)
34
35
36 # # #
37 # setup the simulation
38 # calc the orbital period
39 T = 2 * np.pi * np.power(a, 1.5) / np.sqrt(params['mu_E'])
40 dt = 10.0 # [sec]
41 n orbits = 15.0
42 tspan = np.arange(0.0, n_orbits*T, dt)
43
44 \times 0 = np.concatenate((x 1, x 2))
45
46 # solve
47 x sol = integrate.odeint(func=kepler J2 ODE.kepler J2 ODE,
   t=tspan, y0=x 0, tfirst=True, args=(params, ), rtol=1.0e-12
```

```
File - /Users/iancooke/Dropbox/CUBoulder/Grad Year 2/FormationFlying/Homeworks_PyCharm/HW1/Prob8.py
47 , atol=1.0e-12)
48 print(x_sol.shape)
49 print(tspan[-1])
50
51 # plot first
52 font = {'family' : 'arial',
            'weight' : 'normal',
53
                    : 12}
54
            'size'
55 fig1 = plt.figure(1, figsize=(9, 4.5))
56 plt.rc('font', **font)
57 ax = fig1.gca(projection='3d')
58 ax.plot(x_sol[:, 0], x_sol[:, 1], x_sol[:, 2], linewidth=0.
   5)
59 ax.scatter(0.0, 0.0, 0.0, s=50, color='green')
60 ax.set_xlabel('X [km]')
61 ax.set_ylabel('Y [km]')
62 ax.set_zlabel('Z [km]')
63 #ax.set title('First Satellite 3D Position')
64 plt.title('Satellite 1 Trajectory')
65 plt.legend(('Orbit', 'Earth'))
66 plt.show()
67
68
69 # plot second
70 fig2 = plt.figure(2, figsize=(9, 4.5))
71 ax = fig2.gca(projection='3d')
72 plt.rc('font', **font)
73 ax.plot(x_sol[:, 6], x_sol[:, 7], x_sol[:, 8], linewidth=0.
74 ax.scatter(0.0, 0.0, 0.0, s=50, color='green')
75 ax.set_xlabel('X [km]')
76 ax.set_ylabel('Y [km]')
77 ax.set_zlabel('Z [km]')
78 #ax.set_title('Second Satellite 3D Position')
79 plt.title('Satellite 2 Trajectory')
80 plt.legend(('Orbit', 'Earth'))
81 plt.show()
82
83 # calc & plot angular momentum and energy
84 momentum = np.zeros((tspan.shape[0], params['n objs']))
85 energy = np.zeros((tspan.shape[0], params['n_objs']))
86 for i in range(tspan_shape[0]):
        for j in range(params['n objs']):
87
            r = x_sol[i, 6*j:3+6*j]
88
            v = x_sol[i, 3+6*j:6+6*j]
89
            momentum[i, j] = np.linalg.norm(np.cross(r, v))
90
91
            energy[i, j] = np.power(np.linalg.norm(v), 2.0) / 2
   .0 - params['mu_E'] / np.linalg.norm(r)
```

93 # print out the momentum at the right times

92

```
File - /Users/iancooke/Dropbox/CUBoulder/Grad Year 2/FormationFlying/Homeworks_PyCharm/HW1/Prob8.py
```

```
94 n_steps = np.shape(np.arange(0.0, T, dt))[0]
95 for i in range(int(n_orbits)):
96
        for j in range(params['n objs']):
            r_{vec} = x_{sol}[i*n_{steps}, 6*j:3+6*j]
97
            v_{vec} = x_{sol}[i*n_{steps}, 3 + 6 * j:6 + 6 * j]
 98
99
            r = np.linalg.norm(r_vec)
100
            v = np.linalg.norm(v_vec)
101
            print(r)
102
            print(v)
103
            h_vec = np.cross(r_vec, v_vec)
            h = np.linalg.norm(h_vec)
104
105
            energy = np_power(v, 2.0) / 2.0 - params['mu_E'] /
     r
106
            #print(h)
107
            #print(energy)
108
```