
Table of Contents

.....	1
Part a	1
Part b	1
Part c	2
Part d	5

```
clear; clc; close all;
```

```
% ASEN 5050 - HW 8 - Problem 2  
% Fall 2024  
% Jash Bhalavat
```

```
% Given  
mu_earth = 3.986004415e5; % km3/s2
```

```
% @ t0 in GCRF  
R0 = [2489.63813, -3916.07418, -5679.05524]; % km  
V0 = [9.13452, -1.91212, 2.57306]; % km/s
```

Part a

Calculate ϵ s and h at t_0

```
[eps_0, h0] = rv2eps_h(R0, V0, mu_earth);
```

Part b

```
theta_star_1 = pi;
```

```
% Get E0  
coe_0 = rv2coe(R0, V0, mu_earth);  
theta_star_0 = coe_0(6);  
e_0 = coe_0(2);  
E_0 = 2 * atan(sqrt((1-e_0)/(1+e_0)) * tan(theta_star_0/2));
```

```
% Calculate X_ref using fg f'ns  
X_ref = fg(R0, V0, theta_star_1, mu_earth);  
R1 = X_ref(1,:);  
V1 = X_ref(2,:);
```

```
% Get E1  
coe_1 = rv2coe(R1, V1, mu_earth);  
e_1 = coe_1(2);  
E_1 = 2 * atan(sqrt((1-e_1)/(1+e_1)) * tan(theta_star_1/2));
```

```
% Calculate time from t0 to t1  
a = coe_0(1);
```

```
P = 2*pi*sqrt(a^3/mu_earth);
t0_to_t1 = P/(2*pi) * ((E_1 - e_1*sin(E_1)) - (E_0 - e_1*sin(E_0)));
```

Part c

Initial State

```
state0 = [R0'; V0'];

% Set options for ODE45
options = odeset('RelTol', 1e-8, 'AbsTol', 1e-8);

% Call ODE45 function
[tout, xout] = ode45(@(t, state)x_2bp(t, state, mu_earth), [0 t0_to_t1],
state0, options);

% Calculate eps and h for all points
for i = 1:length(tout)
    [eps(i), h(i)] = rv2eps_h(xout(i, 1:3), xout(i, 4:6), mu_earth);
end

% Plot 3D trajectory
figure()
plot3(xout(:,1), xout(:,2), xout(:,3))
hold on
plot3(0, 0, 0, '.', 'MarkerSize',100, 'Color', 'blue')
plot3(R0(1), R0(2), R0(3), '.', 'MarkerSize', 10)
hold off
legend("Trajectory", "Earth", "Initial Condition")
title("3D Trajectory computed using ODE45")
xlabel("X [km]")
ylabel("Y [km]")
zlabel("Z [km]")

% Plot Specific Energy and Angular Momentum Magnitude
figure()
plot(tout, eps, 'LineWidth', 2)
ylim([eps_0-5, eps_0+5])
title("Specific Energy Time Evolution")
xlabel("Time [sec]")
ylabel("Specific Energy [km^2/s^2]")

figure()
plot(tout, h, 'LineWidth', 2)
ylim([h0-5000, h0+5000])
title("Time Evolution of Specific Angular Momentum Magnitude")
xlabel("Time [sec]")
ylabel("Specific Angular Momentum Magnitude [km^2/s]")

function statedot = x_2bp(t, state, mu_earth)
    % 2 Body Problem Function to pass to ODE45
    x = state(1);
    y = state(2);
```

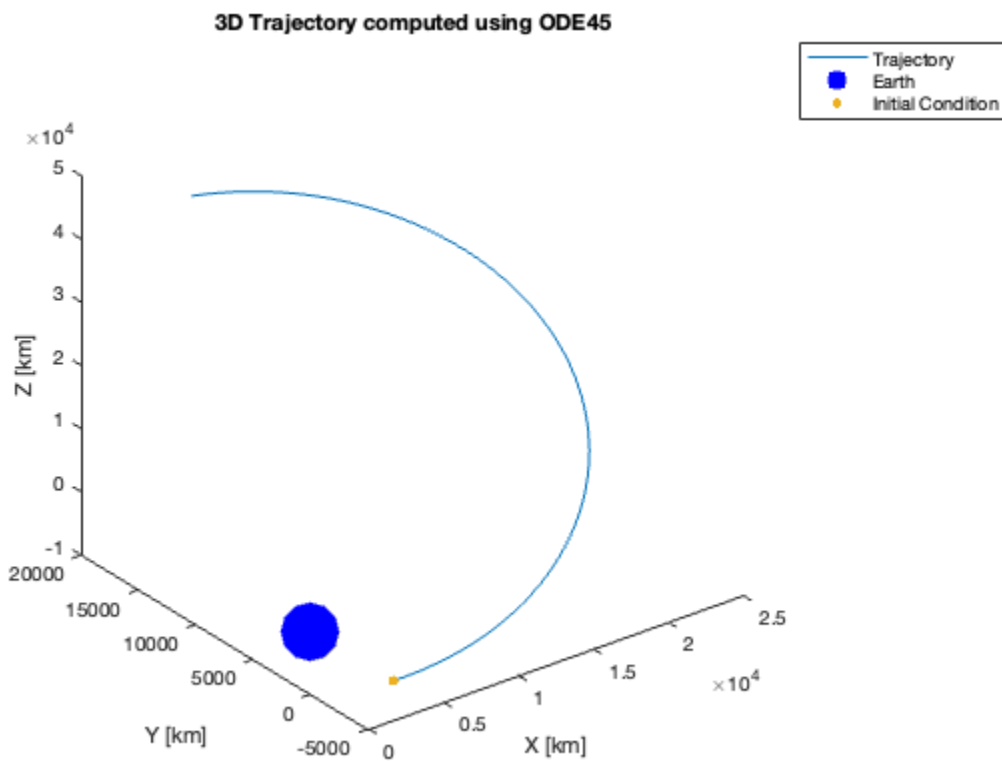
```

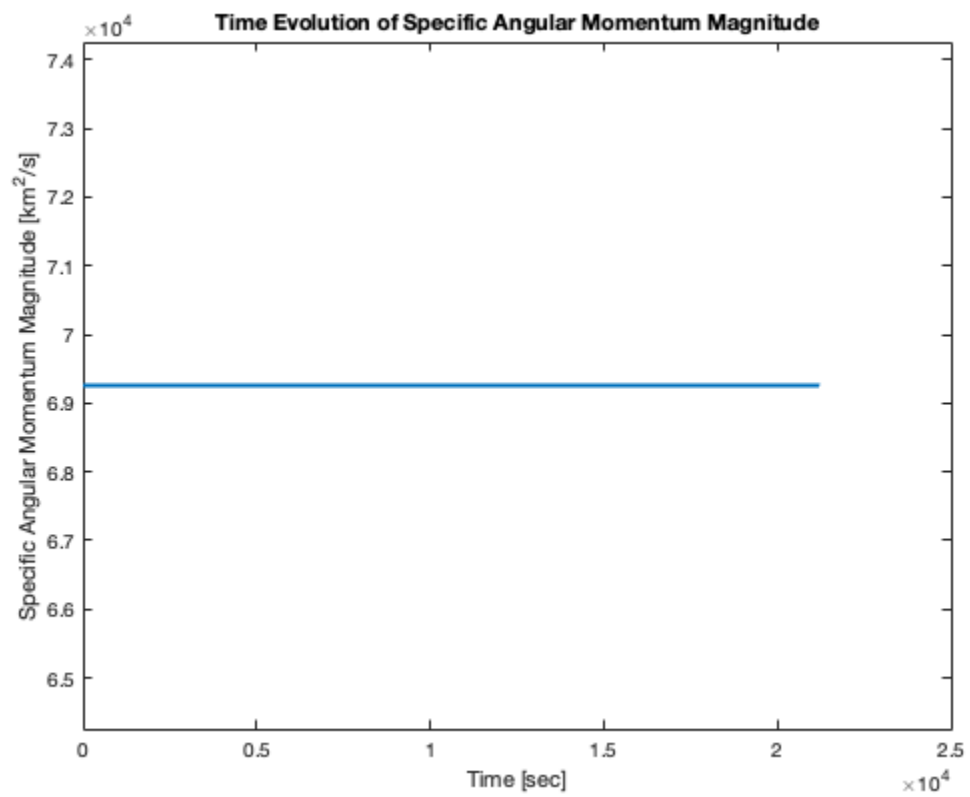
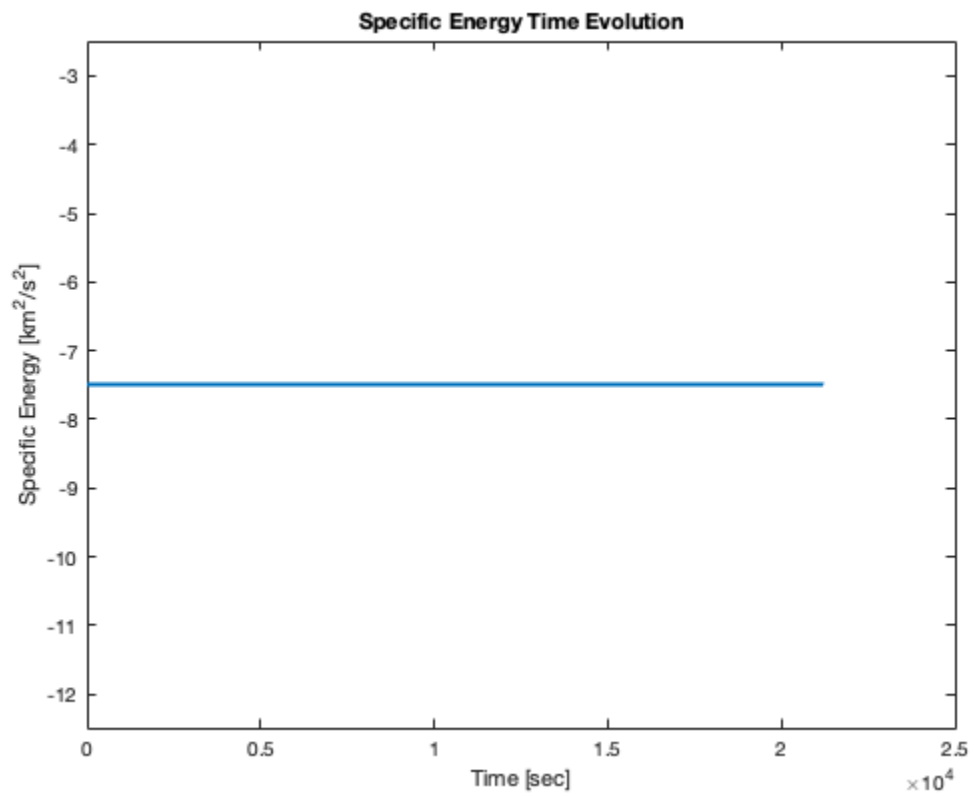
z = state(3);
xdot = state(4);
ydot = state(5);
zdot = state(6);

% r is magnitude of r_vec
r = sqrt(x^2 + y^2 + z^2);

% Derivative output is 2 body problem without any perturbations
statedot(1, 1) = xdot;
statedot(2, 1) = ydot;
statedot(3, 1) = zdot;
statedot(4, 1) = -mu_earth/r^3 * x;
statedot(5, 1) = -mu_earth/r^3 * y;
statedot(6, 1) = -mu_earth/r^3 * z;
end

```





Part d

```
abs_rel_tols = [1e-4, 1e-6, 1e-8, 1e-10, 1e-12];

ref_R1_mag = norm(R1);
ref_V1_mag = norm(V1);

for i = 1:length(abs_rel_tols)
    options = odeset('RelTol', abs_rel_tols(i), 'AbsTol', abs_rel_tols(i));

    tic;
    [tout, xout] = ode45(@(t, state)x_2bp(t, state, mu_earth), [0 t0_to_t1],
state0, options);
    time_taken(i) = toc;

    ode_R1_mag = norm(xout(end,1:3));
    ode_V1_mag = norm(xout(end,4:6));

    [ode_eps_1, ode_h_1] = rv2eps_h(xout(end, 1:3), xout(end, 4:6), mu_earth);

    delta_R(i) = abs(ode_R1_mag - ref_R1_mag);
    delta_V(i) = abs(ode_V1_mag - ref_V1_mag);

    delta_eps(i) = abs(eps_0 - ode_eps_1);
    delta_h(i) = abs(h0 - ode_h_1);
end
```

```
% Part f
```

```
% Perturbations Overview, Slide 24
% Inclination is 63.4 deg, and AOP is 270 deg
% So, RAAN_dot is less than 0 and line of nodes shifts west, orbit is
% prograde
% Since inclination is 63.4 deg, the periapsis is stationary
```

Published with MATLAB® R2024a