ECE 6390: Homework 1

$\label{eq:conjaction} \begin{aligned} & \text{Hyeonjae Park} \\ & \textit{Georgia Institute of Technology} \end{aligned}$

August 31, 2025

1 Result.

Case	Body	Type	a [km]	e	\overline{T}
a	Earth	Elliptic	26823.688	0.25439	12.14 h
b	Earth	Elliptic	68020.478	0.79446	49.04 h
$^{\mathrm{c}}$	Earth	Open (hyperbola)	_	1.52661	
d	Sun	Elliptic	2667330034.581 (17.83 au)	0.96700	$75.29~\mathrm{yr}$

Table 1: Summary of simulated orbits (cases a-d).

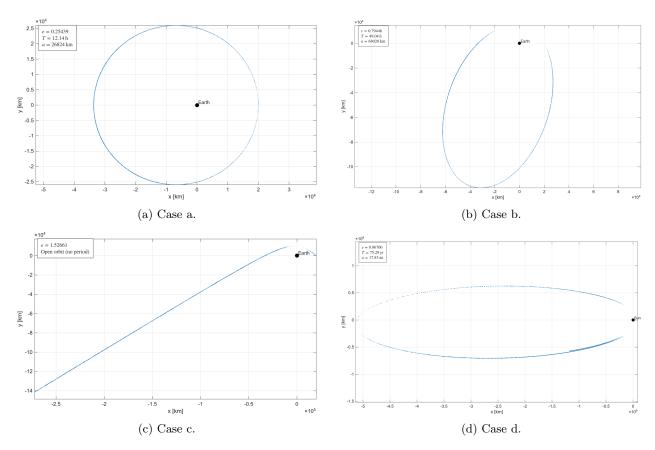


Figure 1: Orbits generated from the initial conditions (cases a-d).

2 Code.

Case a

```
1 function a
3 mu = 398600; % Earth's GM
4 dt = 5; % time step [s]
6 % Initial values
7 r = 20000: \% [km]
8 theta = 0; % [rad]
9 V_r0 = 0.0; % [km/s]
10 V_theta0 = 5.0; % [km/s]
12 % Reset delta
13 delta_r = V_r0 * dt; % [km]
14 delta_theta = (V_theta0 / r) * dt; % [rad]
16 % Simulation Time & History
v^2 = V_r^2 + V_theta^2; % speed^2 [km^2/s^2]
18 eps = 0.5 * v2 - mu/r; % energy
19 h = r * V_theta0; % angular momentum
e = sqrt(1 + (2*eps*h^2)/(mu^2));
21
22 if eps < 0 && e < 1
      a = -mu/(2*eps);
       T = 2*pi*sqrt(a^3/mu);
24
      fprintf('Elliptic orbit: a=%.3f km, e=%.5f, T=%.1f s (%.2f h)\n', a, e, T, T/3600);
25
26 else
       a = NaN; T = NaN;
27
       fprintf('Open orbit: e=%.5f (no period)\n', e);
29 end
30
31 orbits_to_draw = 1;
32 if ~isnan(T)
      sim_time = orbits_to_draw * T;
34 else
35
       sim_time = 60000;
з6 end
n_steps = ceil(sim_time / dt);
39 r_history = zeros(n_steps+1, 1);
40 theta_history = zeros(n_steps+1, 1);
<sub>41</sub> r_history(1) = r;
42 theta_history(1) = theta;
43
44 for n = 1:n_steps
45
       % Position update
       r_next = r + delta_r;
46
       theta_next = theta + delta_theta;
47
48
49
       % Delta update
50
       r_mid = r + 0.5*delta_r;
       delta_r_next = delta_r + (r_mid*(delta_theta^2) - (mu/(r^2))*dt^2);
51
      delta_theta_next = delta_theta - (2*delta_r*delta_theta / r_mid);
52
53
       % Status update
54
55
       r = r_next;
       theta = theta_next;
56
       delta_r = delta_r_next;
       delta_theta = delta_theta_next;
58
59
60
       % Save
       r_history(n+1) = r;
61
       theta_history(n+1) = theta;
63 end
```

```
64
65 x = r_history.*cos(theta_history);
66 y = r_history.*sin(theta_history);
68 figure('Color', 'w');
69 plot(x, y, 'LineWidth', 1.4);
70 hold on
plot(0, 0, 'ko', 'MarkerFaceColor','k', 'MarkerSize', 7);
text(0, 0, 'Earth', 'VerticalAlignment', 'bottom', 'FontSize', 9);
73 axis equal; grid on;
74 xlabel('x [km]'); ylabel('y [km]');
76 ax = gca;
78 if ~isnan(T)
79
       infoLines = { ...
           sprintf('$e=%.5f$', e), ...
80
           sprintf('$T=%.2f\\,\\mathrm{h}$', T/3600), ...
81
           sprintf('$a=%.0f\\,\\mathrm{km}$', a)};
83 else
84
       infoLines = { ...
           sprintf('$e=%.5f$', e), ...
85
            'Open orbit (no period)'};
86
87 end
88
89 text(ax, 0.02, 0.98, infoLines, ...
       'Units','normalized', ...
90
       'Interpreter','latex', ...
'HorizontalAlignment','left', ...
91
92
       'VerticalAlignment','top', ...
93
       'BackgroundColor','w', ...
94
       'EdgeColor','k', ...
95
       'Margin',6, ...
       'FontName', 'Times', 'FontSize', 10);
97
98
99 end
```

Case b

```
1 function b
з mu = 398600; % Earth's GM
4 dt = 10; % time step [s] (increased from 5 to 10 for better visual)
6 % Initial values
7 r = 20000; % [km]
8 theta = 0; % [rad]
9 V_r0 = -3.0; % [km/s]
10 V_theta0 = 5.0; % [km/s]
12 % Reset delta
13 delta_r = V_r0 * dt; % [km]
delta_theta = (V_theta0 / r) * dt; % [rad]
16 % Simulation Time & History
17 v2 = V_r0^2 + V_theta0^2; % speed^2 [km^2/s^2]
18 eps = 0.5*v2 - mu/r; % energy
19 h = r * V_theta0; % angular momentum
e = sqrt(1 + (2*eps*h^2)/(mu^2));
21
22 if eps < 0 && e < 1
      a = -mu/(2*eps);
23
       T = 2*pi*sqrt(a^3/mu);
      fprintf('Elliptic orbit: a=%.3f km, e=%.5f, T=%.1f s (%.2f h)\n', a, e, T, T/3600);
26 else
     a = NaN; T = NaN;
```

```
28
       fprintf('Open orbit: e=%.5f (no period)\n', e);
29 end
30
31 orbits_to_draw = 1;
32 if ~isnan(T)
       sim_time = orbits_to_draw * T;
33
34 else
       sim_time = 60000;
35
з6 end
n_steps = ceil(sim_time / dt);
39 r_history = zeros(n_steps+1, 1);
theta_history = zeros(n_steps+1, 1);
41 r_history(1) = r;
42 theta_history(1) = theta;
44 for n = 1:n_steps
      % Position update
45
46
       r_next = r + delta_r;
       theta_next = theta + delta_theta;
47
       % Delta update
49
       r_mid = r + 0.5*delta_r;
50
       delta_r_next = delta_r + (r_mid*(delta_theta^2) - (mu/(r^2))*dt^2);
51
52
       delta_theta_next = delta_theta - (2*delta_r*delta_theta / r_mid);
53
       % Status update
54
       r = r_next;
55
       theta = theta_next;
56
       delta_r = delta_r_next;
57
58
       delta_theta = delta_theta_next;
59
       % Save
       r_history(n+1) = r;
61
        theta_history(n+1) = theta;
62
63 end
64
65 x = r_history.*cos(theta_history);
66 y = r_history.*sin(theta_history);
68 figure('Color', 'w');
69 plot(x, y, 'LineWidth', 1.4);
70 hold on
plot(0, 0, 'ko', 'MarkerFaceColor','k', 'MarkerSize', 7);
text(0, 0, 'Earth', 'VerticalAlignment','bottom', 'FontSize', 9);
73 axis equal; grid on;
74 xlabel('x [km]'); ylabel('y [km]');
76 ax = gca:
78 if ~isnan(T)
       infoLines = { ...
79
           sprintf('$e=%.5f$', e), ...
80
            sprintf('$T=%.2f\\,\\mathrm{h}$', T/3600), ...
81
            sprintf('$a=%.0f\\,\\mathrm{km}$', a)};
82
   else
83
        infoLines = { ...
           sprintf('$e=%.5f$', e), ...
85
            'Open orbit (no period)'};
86
87 end
88
89 text(ax, 0.02, 0.98, infoLines, ...
       'Units','normalized', ...
90
       'Interpreter','latex', ...
'HorizontalAlignment','left', ...
91
92
       'VerticalAlignment','top', ...
93
       'BackgroundColor','w', ...
   'EdgeColor','k', ...
```

```
96 'Margin',6, ...
97 'FontName','Times', 'FontSize',10);
98
99 end
```

Case c

```
1 function c
2
з mu = 398600; % Earth's GM
4 dt = 5; % time step [s]
6 % Initial values
7 r = 20000; % [km]
 8 theta = 0; % [rad]
9 V_r0 = -6.0; % [km/s]
10 V_theta0 = 5.0; % [km/s]
11
12 % Reset delta
13 delta_r = V_r0 * dt; % [km]
delta_theta = (V_theta0 / r) * dt; % [rad]
16 % Simulation Time & History
v2 = V_r0^2 + V_theta0^2; % speed^2 [km^2/s^2]
18 eps = 0.5*v2 - mu/r; % energy
19 h = r * V_theta0; % angular momentum
20 e = sqrt(1 + (2*eps*h^2)/(mu^2));
21
22 if eps < 0 && e < 1
23
       a = -mu/(2*eps);
       T = 2*pi*sqrt(a^3/mu);
24
       fprintf('Elliptic orbit: a=%.3f km, e=%.5f, T=%.1f s (%.2f h)\n', a, e, T, T/3600);
25
26 else
       a = NaN; T = NaN;
27
       fprintf('Open orbit: e=%.5f (no period)\n', e);
28
29 end
30
31 orbits_to_draw = 1;
32 if ~isnan(T)
33
       sim_time = orbits_to_draw * T;
34 else
35
       sim_time = 60000;
з6 end
n_steps = ceil(sim_time / dt);
38
39 r_history = zeros(n_steps+1, 1);
40 theta_history = zeros(n_steps+1, 1);
r_{\text{history}}(1) = r;
theta_history(1) = theta;
43
44
   for n = 1:n\_steps
       % Position update
45
       r_next = r + delta_r;
46
47
       theta_next = theta + delta_theta;
48
49
       % Delta update
       r_mid = r + 0.5*delta_r;
50
51
       delta_r_next = delta_r + (r_mid*(delta_theta^2) - (mu/(r^2))*dt^2);
52
       delta_theta_next = delta_theta - (2*delta_r*delta_theta / r_mid);
53
54
       % Status update
       r = r_next;
55
56
       theta = theta_next;
       delta_r = delta_r_next;
58
       delta_theta = delta_theta_next;
59
```

```
60 % Save
61
       r_history(n+1) = r;
       theta_history(n+1) = theta;
62
63 end
64
65 x = r_history.*cos(theta_history);
66 y = r_history.*sin(theta_history);
67
68 figure('Color', 'w');
69 plot(x, y, 'LineWidth', 1.4);
70 hold on
plot(0, 0, 'ko', 'MarkerFaceColor', 'k', 'MarkerSize', 7);
text(0, 0, 'Earth', 'VerticalAlignment', 'bottom', 'FontSize', 9);
73 axis equal; grid on;
74 xlabel('x [km]'); ylabel('y [km]');
76 ax = gca;
77
78 if ~isnan(T)
       infoLines = { ...
79
80
           sprintf('$e=%.5f$', e), ...
           sprintf('$T=%.2f\\,\\mathrm{h}$', T/3600), ...
81
           sprintf('$a=%.0f\\,\\mathrm{km}$', a)};
82
83 else
       infoLines = { ...
84
85
           sprintf('$e=%.5f$', e), ...
            'Open orbit (no period)'};
86
87 end
88
   text(ax, 0.02, 0.98, infoLines, ...
89
       'Units','normalized', ...
90
       'Interpreter','latex', ..
91
       'HorizontalAlignment','left', ...
       'VerticalAlignment','top', ...
93
       'BackgroundColor','w', ...
94
       'EdgeColor','k', ...
95
       'Margin',6, ...
96
       'FontName', 'Times', 'FontSize', 10);
97
98
99 end
```

Case d

```
1 function d
2
3 mu = 1.327124400e11; % Sun's GM
4 dt = 21600; % time step [s]
6 % Initial values
7 AU = 149597870.7; % 1 au [km]
8 a_au = 17.83; % semi-major axis [au]
9 e_use = 0.967; % eccentricity
a_{m} = a_{m} * AU;
11
r = a_k m * (1 - e_u se); % perihelion distance [km]
13 theta = 0; % start angle
14 V_r0 = 0.0;
V_{theta0} = -sqrt( mu*(1+e_use) / (a_km*(1 - e_use)) );
17 % Reset delta
18 delta_r = V_r0 * dt; % [km]
delta_theta = (V_theta0 / r) * dt; % [rad]
21 % Simulation Time & History
v2 = V_r0^2 + V_theta0^2; % speed^2 [km^2/s^2]
23 eps = 0.5*v2 - mu/r; % energy
```

```
h = r * V_theta0; % angular momentum
e = sqrt(1 + (2*eps*h^2)/(mu^2));
26
27 if eps < 0 && e < 1
28
       a = -mu/(2*eps);
29
       T = 2*pi*sqrt(a^3/mu);
30
       T_{yr} = T / (365.25*86400);
       fprintf('Elliptic orbit: a=%.3f km, e=%.5f, T=%.2f yr\n', a, e, T_yr);
31
32 else
       a = NaN: T = NaN:
33
34
        fprintf('Open orbit: e=%.5f (no period)\n', e);
35 end
36
37 orbits_to_draw = 1;
38 if ~isnan(T)
       sim_time = orbits_to_draw * T;
39
40 else
       sim_time = 60000;
41
42 end
43 n_steps = ceil(sim_time / dt);
r_history = zeros(n_steps+1, 1);
theta_history = zeros(n_steps+1, 1);
r_{\text{history}}(1) = r;
48 theta_history(1) = theta;
49
50 for n = 1:n_steps
       % Position update
51
       r_next = r + delta_r;
52
       theta_next = theta + delta_theta;
53
54
       % Delta update
55
       r_mid = r + 0.5*delta_r;
56
       delta\_r\_next = delta\_r + (r\_mid*(delta\_theta^2) - (mu/(r^2))*dt^2);
57
       delta_theta_next = delta_theta - (2*delta_r*delta_theta / r_mid);
58
59
       % Status update
60
61
       r = r_next;
       theta = theta next:
62
63
       delta_r = delta_r_next;
       delta_theta = delta_theta_next;
64
65
       % Save
       r_history(n+1) = r;
67
        theta_history(n+1) = theta;
68
69 end
70
71 x = r_history.*cos(theta_history);
72 y = r_history.*sin(theta_history);
74 figure('Color', 'w');
75 plot(x, y, 'LineWidth', 1.4);
76 hold on
77 plot(0, 0, 'ko', 'MarkerFaceColor','k', 'MarkerSize', 7);
78 text(0, 0, 'Sun', 'VerticalAlignment','bottom', 'FontSize', 9);
79 axis equal; grid on;
so xlabel('x [km]'); ylabel('y [km]');
81
82 ax = gca;
83
84 if ~isnan(T)
       infoLines = { ...
           sprintf('$e=%.5f$', e), ...
86
            sprintf('$T=\%.2f\\,\mathrm{yr}$', T_yr), \dots
87
88
            sprintf('$a=%.2f\\,\\mathrm{au}$', a/AU) };
89 else
       infoLines = { ...
   sprintf('$e=%.5f$', e), ...
```

```
'Open orbit (no period)'};
93 end
94
95 text(ax, 0.02, 0.98, infoLines, ...
           'Units', 'normalized', ...
'Interpreter', 'latex', ...
'HorizontalAlignment', 'left', ...
'VerticalAlignment', 'top', ...
'BackgroundColor', 'w', ...
96
97
98
99
100
             'EdgeColor','k', ...
'Margin',6, ...
'FontName','Times', 'FontSize',10);
101
102
103
104
105 end
```