ECE 6390: Homework 1

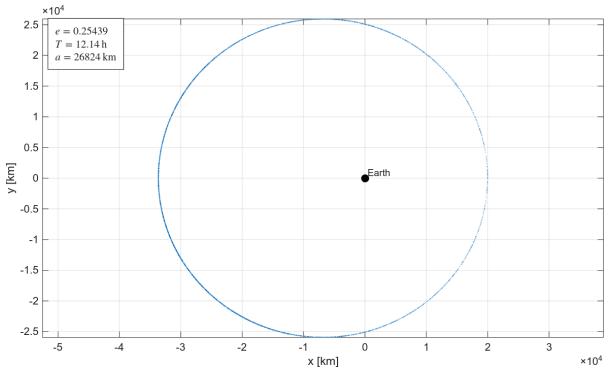
Hyeonjae Park
Georgia Institute of Technology

August 31, 2025

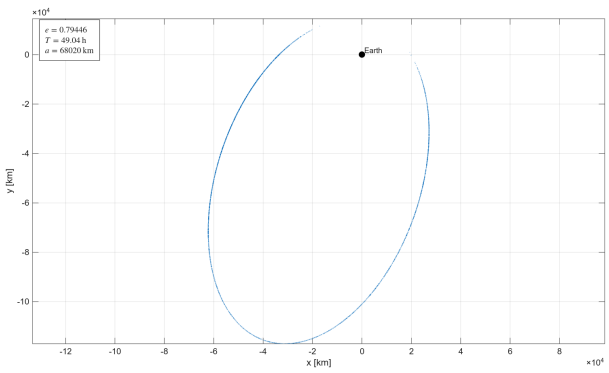
1 Result.

Case	Body	Type	a [km]	e	T
a	Earth	Elliptic	26823.688	0.25439	12.14 h
b	Earth	Elliptic	68020.478	0.79446	49.04 h
c	Earth	Open (hyperbola)	—	1.52661	—
d	Sun	Elliptic	2667330034.581 (17.83 au)	0.96700	75.29 yr

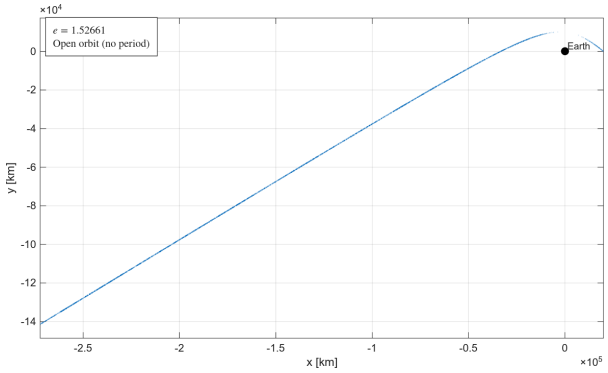
Table 1: Summary of simulated orbits (cases a–d).



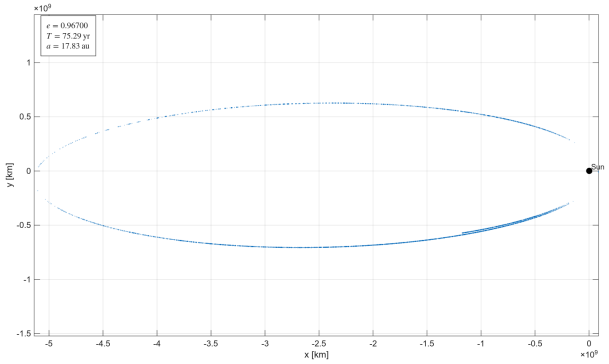
(a) Case a.



(b) Case b.



(c) Case c.



(d) Case d.

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

2 Code.

Case a

```
function a

mu = 398600; % Earth's GM
dt = 5; % time step [s]

% Initial values
r = 20000; % [km]
theta = 0; % [rad]
V_r0 = 0.0; % [km/s]
V_theta0 = 5.0; % [km/s]

% Reset delta
delta_r = V_r0 * dt; % [km]
delta_theta = (V_theta0 / r) * dt; % [rad]

% Simulation Time & History
v2 = V_r0^2 + V_theta0^2; % speed^2 [km^2/s^2]
eps = 0.5*v2 - mu/r; % energy
h  = r * V_theta0; % angular momentum
e  = sqrt(1 + (2*eps*h^2)/(mu^2));

if eps < 0 && e < 1
    a = -mu/(2*eps);
    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);
else
    a = NaN; T = NaN;
    fprintf('Open orbit: e=%.5f (no period)\n', e);
end

orbits_to_draw = 1;
if ~isnan(T)
    sim_time = orbits_to_draw * T;
else
    sim_time = 60000;
end
n_steps = ceil(sim_time / dt);

r_history = zeros(n_steps+1, 1);
theta_history = zeros(n_steps+1, 1);
r_history(1) = r;
theta_history(1) = theta;

for n = 1:n_steps
    % Position update
    r_next = r + delta_r;
    theta_next = theta + delta_theta;

    % Delta update
    r_mid = r + 0.5*delta_r;
    delta_r_next = delta_r + (r_mid*(delta_theta^2) - (mu/(r^2))*dt^2);
    delta_theta_next = delta_theta - (2*delta_r*delta_theta / r_mid);

    % Status update
    r = r_next;
    theta = theta_next;
    delta_r = delta_r_next;
    delta_theta = delta_theta_next;

    % Save
    r_history(n+1) = r;
    theta_history(n+1) = theta;
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
71  plot(0, 0, 'ko', 'MarkerFaceColor','k', 'MarkerSize', 7);
72  text(0, 0, ' Earth', 'VerticalAlignment','bottom', 'FontSize', 9);
73  axis equal; grid on;
74  xlabel('x [km]'); ylabel('y [km]');
75
76  ax = gca;
77
78  if ~isnan(T)
79      infoLines = { ...
80          sprintf('$e=%.5f$', e), ...
81          sprintf('$T=%.2f\\,\\mathrm{h}$', T/3600), ...
82          sprintf('$a=%.0f\\,\\mathrm{km}$', a)};
83  else
84      infoLines = { ...
85          sprintf('$e=%.5f$', e), ...
86          'Open orbit (no period)'};
87  end
88
89  text(ax, 0.02, 0.98, infoLines, ...
90      'Units','normalized', ...
91      'Interpreter','latex', ...
92      'HorizontalAlignment','left', ...
93      'VerticalAlignment','top', ...
94      'BackgroundColor','w', ...
95      'EdgeColor','k', ...
96      'Margin',6, ...
97      'FontName','Times', 'FontSize',10);
98
99  end
```

Case b

```
1   function b
2
3   mu = 398600; % Earth's GM
4   dt = 10; % time step [s] (increased from 5 to 10 for better visual)
5
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]
11
12  % Reset delta
13  delta_r = V_r0 * dt; % [km]
14  delta_theta = (V_theta0 / r) * dt; % [rad]
15
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
20  e   = sqrt(1 + (2*eps*h^2)/(mu^2));
21
22  if eps < 0 && e < 1
23      a = -mu/(2*eps);
24      T = 2*pi*sqrt(a^3/mu);
25      fprintf('Elliptic orbit: a=%.3f km, e=%.5f, T=%.1f s (%.2f h)\n', a, e, T, T/3600);
26  else
27      a = NaN; T = NaN;
```

```
    fprintf('Open orbit: e=%.5f (no period)\n', e);
end

orbits_to_draw = 1;
if ~isnan(T)
    sim_time = orbits_to_draw * T;
else
    sim_time = 60000;
end
n_steps = ceil(sim_time / dt);

r_history = zeros(n_steps+1, 1);
theta_history = zeros(n_steps+1, 1);
r_history(1) = r;
theta_history(1) = theta;

for n = 1:n_steps
    % Position update
    r_next = r + delta_r;
    theta_next = theta + delta_theta;

    % Delta update
    r_mid = r + 0.5*delta_r;
    delta_r_next = delta_r + (r_mid*(delta_theta^2) - (mu/(r^2))*dt^2);
    delta_theta_next = delta_theta - (2*delta_r*delta_theta / r_mid);

    % Status update
    r = r_next;
    theta = theta_next;
    delta_r = delta_r_next;
    delta_theta = delta_theta_next;

    % Save
    r_history(n+1) = r;
    theta_history(n+1) = theta;
end

x = r_history.*cos(theta_history);
y = r_history.*sin(theta_history);

figure('Color', 'w');
plot(x, y, 'LineWidth', 1.4);
hold on
plot(0, 0, 'ko', 'MarkerFaceColor','k', 'MarkerSize', 7);
text(0, 0, ' Earth', 'VerticalAlignment','bottom', 'FontSize', 9);
axis equal; grid on;
xlabel('x [km]'); ylabel('y [km]');

ax = gca;

if ~isnan(T)
    infoLines = { ...
        sprintf('$e=%.5f$', e), ...
        sprintf('$T=%.2f\\,\\mathrm{h}$', T/3600), ...
        sprintf('$a=%.0f\\,\\mathrm{km}$', a)};
else
    infoLines = { ...
        sprintf('$e=%.5f$', e), ...
        'Open orbit (no period)'};
end

text(ax, 0.02, 0.98, infoLines, ...
    'Units','normalized', ...
    'Interpreter','latex', ...
    'HorizontalAlignment','left', ...
    'VerticalAlignment','top', ...
    'BackgroundColor','w', ...
    'EdgeColor','k', ...
```

```
    'Margin',6, ...
    'FontName','Times', 'FontSize',10);

end
```

Case c

```
function c

mu = 398600; % Earth's GM
dt = 5; % time step [s]

% Initial values
r = 20000; % [km]
theta = 0; % [rad]
V_r0 = -6.0; % [km/s]
V_theta0 = 5.0; % [km/s]

% Reset delta
delta_r = V_r0 * dt; % [km]
delta_theta = (V_theta0 / r) * dt; % [rad]

% Simulation Time & History
v2  = V_r0^2 + V_theta0^2; % speed^2 [km^2/s^2]
eps = 0.5*v2 - mu/r; % energy
h   = r * V_theta0; % angular momentum
e   = sqrt(1 + (2*eps*h^2)/(mu^2));

if eps < 0 && e < 1
    a = -mu/(2*eps);
    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);
else
    a = NaN; T = NaN;
    fprintf('Open orbit: e=%.5f (no period)\n', e);
end

orbits_to_draw = 1;
if ~isnan(T)
    sim_time = orbits_to_draw * T;
else
    sim_time = 60000;
end
n_steps = ceil(sim_time / dt);

r_history = zeros(n_steps+1, 1);
theta_history = zeros(n_steps+1, 1);
r_history(1) = r;
theta_history(1) = theta;

for n = 1:n_steps
    % Position update
    r_next = r + delta_r;
    theta_next = theta + delta_theta;

    % Delta update
    r_mid = r + 0.5*delta_r;
    delta_r_next = delta_r + (r_mid*(delta_theta^2) - (mu/(r^2))*dt^2);
    delta_theta_next = delta_theta - (2*delta_r*delta_theta / r_mid);

    % Status update
    r = r_next;
    theta = theta_next;
    delta_r = delta_r_next;
    delta_theta = delta_theta_next;

```

```
    % Save
    r_history(n+1) = r;
    theta_history(n+1) = theta;
end

x = r_history.*cos(theta_history);
y = r_history.*sin(theta_history);

figure('Color', 'w');
plot(x, y, 'LineWidth', 1.4);
hold on
plot(0, 0, 'ko', 'MarkerFaceColor','k', 'MarkerSize', 7);
text(0, 0, ' Earth', 'VerticalAlignment','bottom', 'FontSize', 9);
axis equal; grid on;
xlabel('x [km]'); ylabel('y [km]');

ax = gca;

if ~isnan(T)
    infoLines = { ...
        sprintf('$e=%.5f$', e), ...
        sprintf('$T=%.2f\\,\\mathrm{h}$', T/3600), ...
        sprintf('$a=%.0f\\,\\mathrm{km}$', a)};
else
    infoLines = { ...
        sprintf('$e=%.5f$', e), ...
        'Open orbit (no period)'};
end

text(ax, 0.02, 0.98, infoLines, ...
    'Units','normalized', ...
    'Interpreter','latex', ...
    'HorizontalAlignment','left', ...
    'VerticalAlignment','top', ...
    'BackgroundColor','w', ...
    'EdgeColor','k', ...
    'Margin',6, ...
    'FontName','Times', 'FontSize',10);

end
```

Case d

```
function d

mu = 1.327124400e11; % Sun's GM
dt = 21600; % time step [s]

% Initial values
AU = 149597870.7; % 1 au [km]
a_au = 17.83; % semi-major axis [au]
e_use = 0.967; % eccentricity
a_km = a_au * AU;

r = a_km * (1 - e_use); % perihelion distance [km]
theta = 0; % start angle
V_r0 = 0.0;
V_theta0 = -sqrt( mu*(1+e_use) / (a_km*(1 - e_use)) );

% Reset delta
delta_r = V_r0 * dt; % [km]
delta_theta = (V_theta0 / r) * dt; % [rad]

% Simulation Time & History
v2 = V_r0^2 + V_theta0^2; % speed^2 [km^2/s^2]
eps = 0.5*v2 - mu/r; % energy
```

```
h = r * V_theta0; % angular momentum
e = sqrt(1 + (2*eps*h^2)/(mu^2));

if eps < 0 && e < 1
    a = -mu/(2*eps);
    T = 2*pi*sqrt(a^3/mu);
    T_yr = T / (365.25*86400);
    fprintf('Elliptic orbit: a=%.3f km, e=%.5f, T=%.2f yr\n', a, e, T_yr);
else
    a = NaN; T = NaN;
    fprintf('Open orbit: e=%.5f (no period)\n', e);
end

orbits_to_draw = 1;
if ~isnan(T)
    sim_time = orbits_to_draw * T;
else
    sim_time = 60000;
end
n_steps = ceil(sim_time / dt);

r_history = zeros(n_steps+1, 1);
theta_history = zeros(n_steps+1, 1);
r_history(1) = r;
theta_history(1) = theta;

for n = 1:n_steps
    % Position update
    r_next = r + delta_r;
    theta_next = theta + delta_theta;

    % Delta update
    r_mid = r + 0.5*delta_r;
    delta_r_next = delta_r + (r_mid*(delta_theta^2) - (mu/(r^2))*dt^2);
    delta_theta_next = delta_theta - (2*delta_r*delta_theta / r_mid);

    % Status update
    r = r_next;
    theta = theta_next;
    delta_r = delta_r_next;
    delta_theta = delta_theta_next;

    % Save
    r_history(n+1) = r;
    theta_history(n+1) = theta;
end

x = r_history.*cos(theta_history);
y = r_history.*sin(theta_history);

figure('Color', 'w');
plot(x, y, 'LineWidth', 1.4);
hold on
plot(0, 0, 'ko', 'MarkerFaceColor','k', 'MarkerSize', 7);
text(0, 0, ' Sun', 'VerticalAlignment','bottom', 'FontSize', 9);
axis equal; grid on;
xlabel('x [km]'); ylabel('y [km]');

ax = gca;

if ~isnan(T)
    infoLines = { ...
        sprintf('$e=%.5f$', e), ...
        sprintf('$T=%.2f\\,\\mathrm{yr}$', T_yr), ...
        sprintf('$a=%.2f\\,\\mathrm{au}$', a/AU) };
else
    infoLines = { ...
        sprintf('$e=%.5f$', e), ...
```

```
        'Open orbit (no period)'};
end

text(ax, 0.02, 0.98, infoLines, ...
    'Units','normalized', ...
    'Interpreter','latex', ...
    'HorizontalAlignment','left', ...
    'VerticalAlignment','top', ...
    'BackgroundColor','w', ...
    'EdgeColor','k', ...
    'Margin',6, ...
    'FontName','Times', 'FontSize',10);

end
```