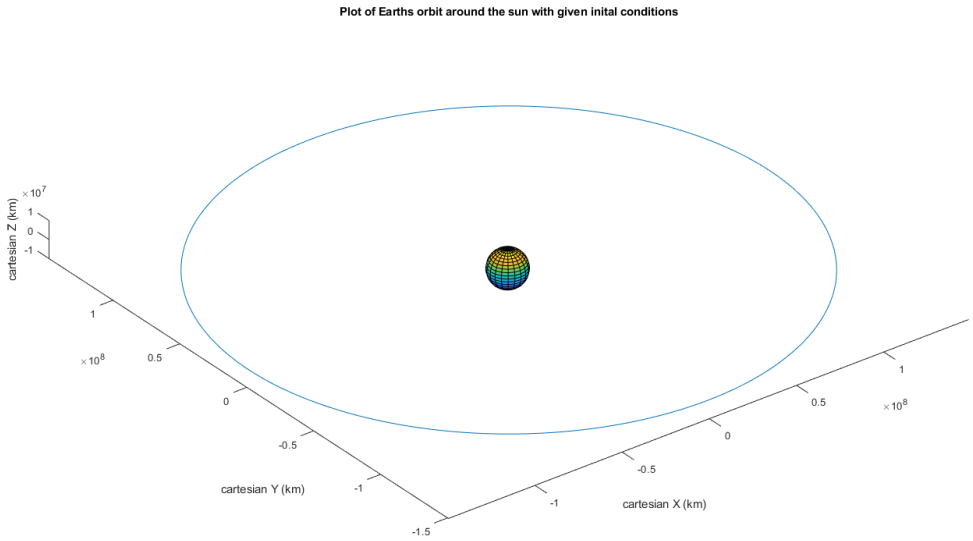
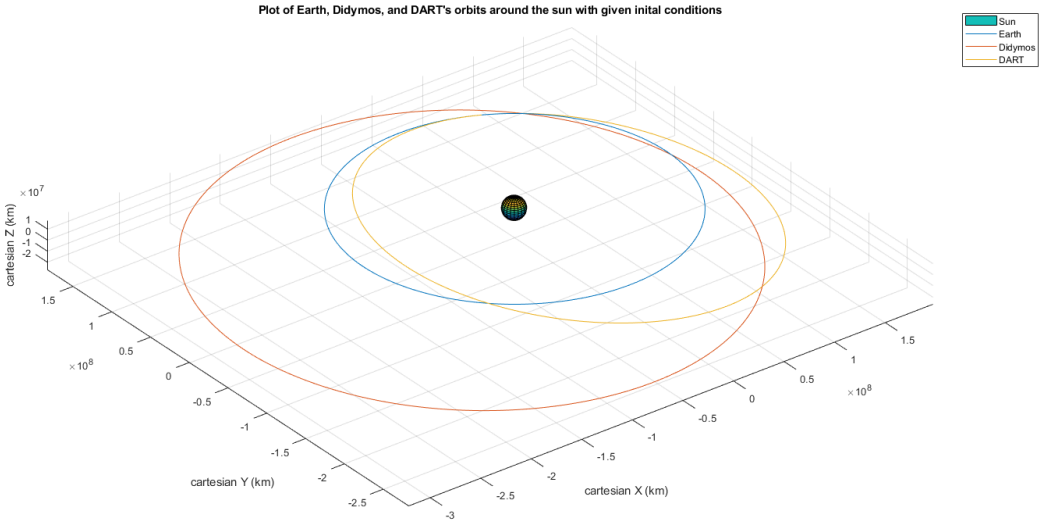
ENAE404 : HW0 Romeo Perlstein, section 0101

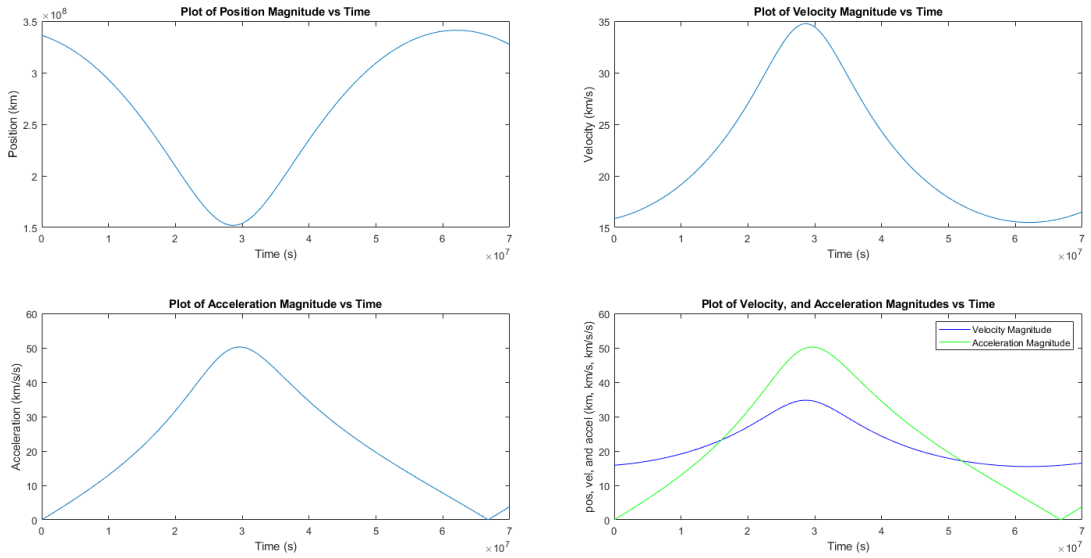
Preliminary Work:



Q1)



Q2)



Q3)

To me, the magnitude plots make sense because the given equation of motion relies on the radius of the body to the central planet (the sun). Since the radius varies within the orbit, you should expect to see variations in a planet's speed and acceleration (and, of course, its position). If the position were thought to behave like a cosine function, then it makes sense that the velocity plot would like a sine function (the derivative of the cosine), or at least that their crests and troughs would be swapped (I think?). But, to summarize, it makes sense that as the Didymos gets closer to the sun, its velocity and acceleration will increase, because it's experiencing a larger gravitational force!

Q4)

Didymos Final State Vector:

| X = -195277874.348987 (km) |

| Y = -262465171.020025 (km) |

| Z = 6558785.74996602 (km) |

| vX = 15.0133881743423 (km/s) |

| vY = -6.82831729701378 (km/s) |

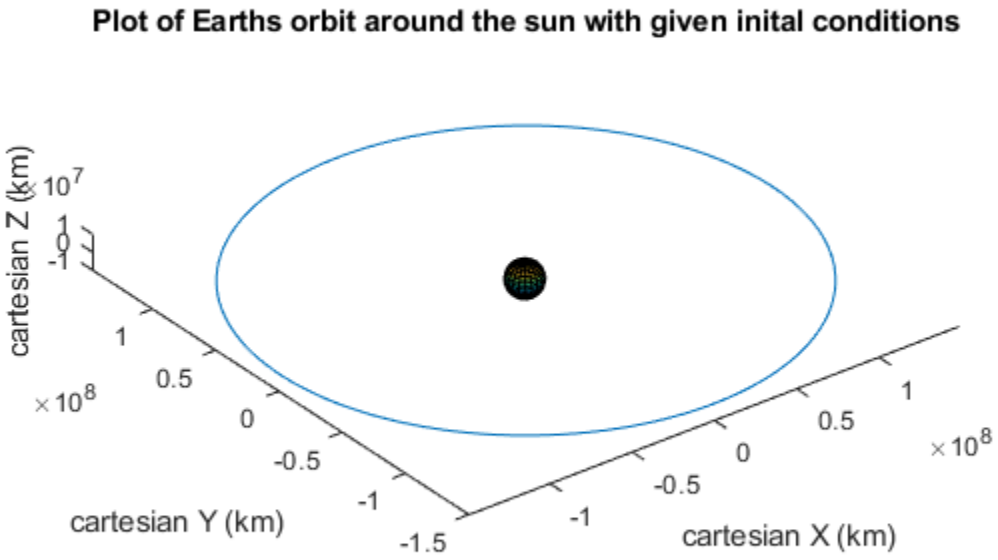
| vZ = -0.972618243181786 (km/s) |

*See attached source code

Table of Contents


```
%-------- HW 0 MATLAB code --------%
% Romeo Perlstein, section 0101 %

% Earth Initial Positions (km)
Earth_initial_x = 6.82500*10^7;
Earth_initial_y = 1.30864*10^8;
Earth_initial_z = 1.81329*10^4;

% Earth Initial Velocities (km/s)
Earth_initial_vx = -2.67639*10;
Earth_initial_vy = 1.38981*10;
Earth_initial_vz = -9.22794*10^-4;

Earth_initial_state = [Earth_initial_x; Earth_initial_y; Earth_initial_z;
 Earth_initial_vx; Earth_initial_vy; Earth_initial_vz];

% Constants
mew_sun = 1.32712 * (10^11);
tall_er_ant = (10^-13);
step_size = 10000;
max_time = 70000000;

% Time step
t = [0:step_size:max_time];

% ODE options
ODE_options = odeset("RelTol", tall_er_ant, "AbsTol", tall_er_ant);

% format the data for 64 bit numeros
format long

% Plug it in to ode45
[T1,Y1] = ode45(@myodefun, t, Earth_initial_state, ODE_options, mew_sun);

plot3(Y1(:,1), Y1(:,2), Y1(:,3));
title("Plot of Earths orbit around the sun with given inital conditions")
xlabel("cartesian X (km)")
ylabel("cartesian Y (km)")
zlabel("cartesian Z (km)")
hold on
% Make the sun, for sh*ts and giggles (pardon my french)
[X_1, Y_1, Z_1] = sphere;
```

```
X_2 = X_1*(10^7);
Y_2 = Y_1*(10^7);
Z_2 = Z_1*(10^7);
surf(X_2, Y_2, Z_2);
axis equal
hold off
```

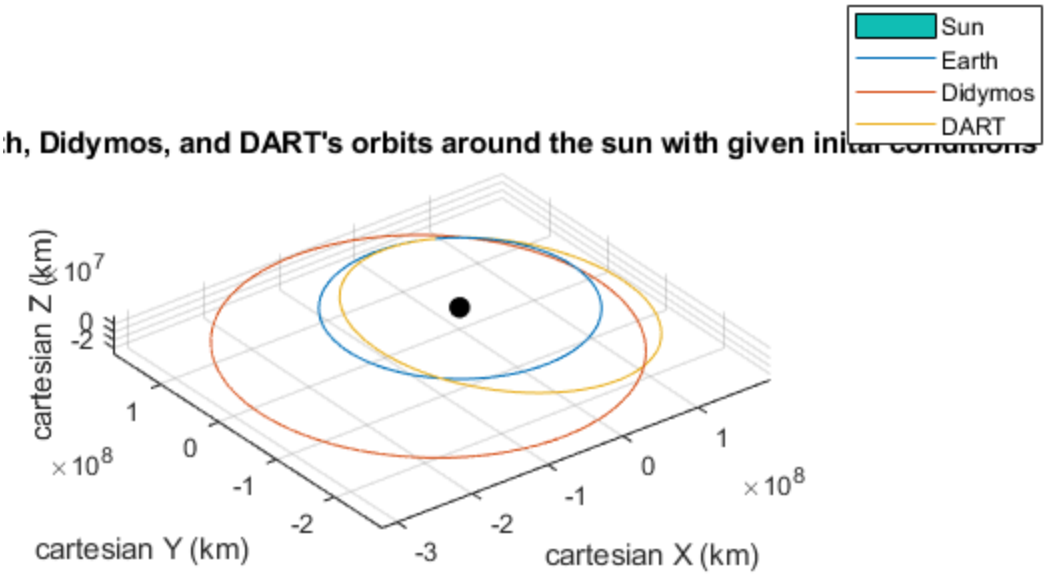


Question 1

Now do it for didymos!

```
didymos_initial_x = -2.39573*10^8;
didymos_initial_y = -2.35661*10^8;
didymos_initial_z = 9.54384*10^6;
didymos_initial_vx = 1.24732*10^1;
didymos_initial_vy = -9.74427*10^0;
didymos_initial_vz = -8.78661*10^-1;
didymos_initial_state = [didymos_initial_x; didymos_initial_y;
 didymos_initial_z; didymos_initial_vx; didymos_initial_vy;
 didymos_initial_vz;0;0;0];

% Now do it for DART!
DART_initial_x = 6.82409*10^7;
DART_initial_y = 1.30854*10^8;
DART_initial_z = 1.52197*10^4;
DART_initial_vx = -3.06997*10^1;
```

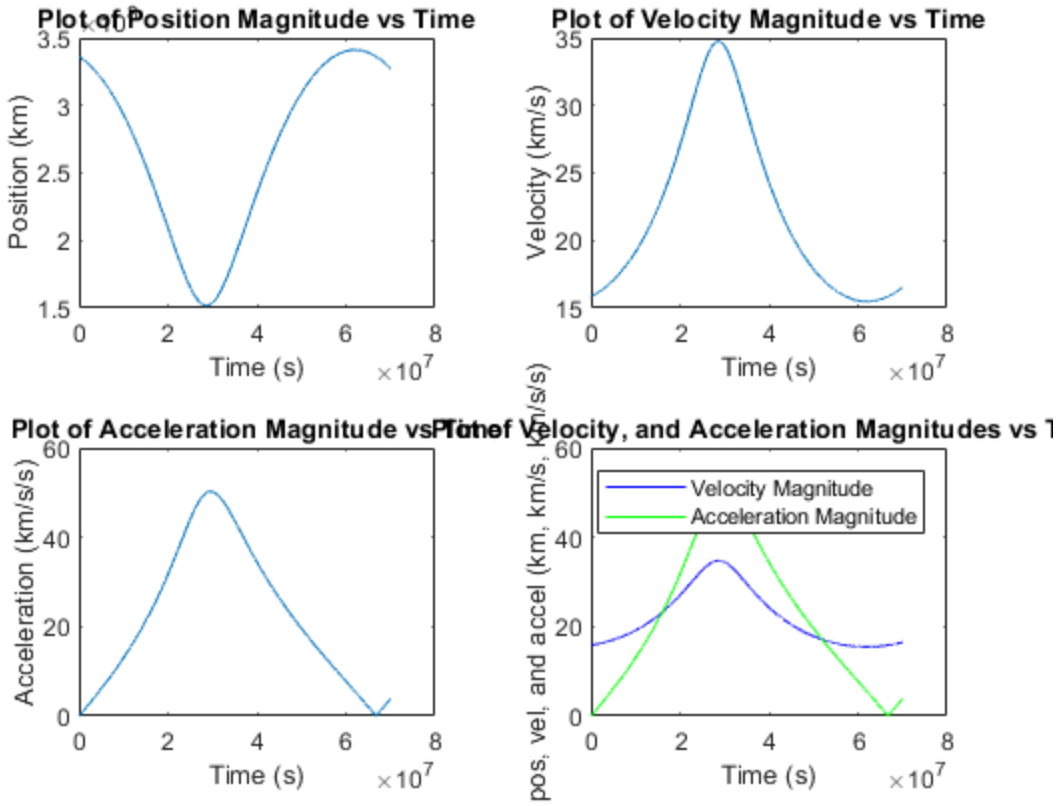
```
DART_initial_vy = 8.11796*10^0;
DART_initial_vz = 3.95772*10^0;
DART_initial_state = [DART_initial_x; DART_initial_y; DART_initial_z;
 DART_initial_vx; DART_initial_vy; DART_initial_vz];

% Get Earth again
[T1_,Y1_] = ode45(@myodefun, t, Earth_initial_state, ODE_options, mew_sun);
% Get didymos data
[T2,Y2] = ode45(@myodefun_mod, t, didymos_initial_state, ODE_options,
 mew_sun);
% Get DART data
[T3,Y3] = ode45(@myodefun, t, DART_initial_state, ODE_options, mew_sun);

figure
% Plot Sun
surf(X_2,Y_2,Z_2)
hold on

title("Plot of Earth, Didymos, and DART's orbits around the sun with given
 inital conditions")
xlabel("cartesian X (km)")
ylabel("cartesian Y (km)")
zlabel("cartesian Z (km)")
% Plot Earth
plot3(Y1_(:,1), Y1_(:,2), Y1_(:,3));
% Plot Didymos
plot3(Y2(:,1), Y2(:,2), Y2(:,3));
% Plot DART
plot3(Y3(:,1), Y3(:,2), Y3(:,3));
legend("Sun", "Earth", "Didymos", "DART")
axis equal
hold off
```



Question 2

```
figure
tiledlayout(2,2);
nexttile % tile1
pos = [Y2(:,1), Y2(:,2), Y2(:,3)];
for i=1:length(Y2(:,1))
    pos_mag(i) = sqrt(pos(i,1)^2+pos(i,2)^2+pos(i,3)^2);
end
plot(T2,pos_mag);
title("Plot of Position Magnitude vs Time")
xlabel("Time (s)")
ylabel("Position (km)")

nexttile % tile 2
vel = [Y2(:,4), Y2(:,5), Y2(:,6)];
for i=1:length(Y2(:,4))
    vel_mag(i) = sqrt(vel(i,1)^2+vel(i,2)^2+vel(i,3)^2);
end
plot(T2,vel_mag);
title("Plot of Velocity Magnitude vs Time")
xlabel("Time (s)")
ylabel("Velocity (km/s)")

nexttile % tile 3
```

```
accel = [Y2(:,7), Y2(:,8), Y2(:,9)];
for i=1:length(Y2(:,7))
    accel_mag(i) = sqrt(accel(i,1)^2+accel(i,2)^2+accel(i,3)^2);
end
plot(T2,accel_mag);
title("Plot of Acceleration Magnitude vs Time")
xlabel("Time (s)")
ylabel("Acceleration (km/s/s)")

nexttile % tile 4
plot(T2, vel_mag, "-b", T2, accel_mag, "-g")
legend("Velocity Magnitude", "Acceleration Magnitude")
title("Plot of Velocity, and Acceleration Magnitudes vs Time")
xlabel("Time (s)")
ylabel("pos, vel, and accel (km, km/s, km/s/s)")
```



Question 3

To me, the magnitude plots make sense because the given equation of motion relies on the radius of the body to the central planet (the sun). Since the radius varies within the orbit, you should expect to see variations in a planets speed and acceleration (and, of course, its position). If the position were though to behave similar to a cosine function, then it makes sense that the velocity plot would like a sine function (the derivative of the cosine), or at least that their crests and troughs would be swapped (I think?). But, to summarize, it makes sense that as the didymos gets closer to the sun, it's velocity and accerlation will increase, because it's experiencing a larger gravitational force!

Question 4

Didymos final state:

```
len = length(Y2(:,1));
final_state = [Y2(len, 1); Y2(len, 2); Y2(len, 3); Y2(len, 4); Y2(len, 5);
 Y2(len, 6)]
```

```
final_state =

   1.0e+08 *

  -1.952778743489875
  -2.624651710200249
   0.065587857499660
   0.000000150133882
  -0.000000068283173
  -0.000000009726182
```

ODE function handles

From ENAE301

```
function ydot = myodefun(t, y, mew)
    r_mag = norm(y(1:3));
    ydot(1,1) = y(4);
    ydot(2,1) = y(5);
    ydot(3,1) = y(6);
    ydot(4,1) = (-mew/r_mag^3)*y(1);
    ydot(5,1) = (-mew/r_mag^3)*y(2);
    ydot(6,1) = (-mew/r_mag^3)*y(3);
end

function ydot = myodefun_mod(t, y, mew)
    r_mag = norm(y(1:3));
    ydot(1,1) = y(4);
    ydot(2,1) = y(5);
    ydot(3,1) = y(6);
    ydot(4,1) = (-mew/r_mag^3)*y(1);
    ydot(5,1) = (-mew/r_mag^3)*y(2);
    ydot(6,1) = (-mew/r_mag^3)*y(3);
    ydot(7,1) = (-mew/r_mag^3)*y(1);
    ydot(8,1) = (-mew/r_mag^3)*y(2);
    ydot(9,1) = (-mew/r_mag^3)*y(3);
end
```

Published with MATLAB® R2022b