

---

## Problem 8

```
clear all; close all; clc

ro = [3207 5459 2714]; %state vectors
vo = [-6.532 .7835 6.142];
ti = 0;
tf = 3600*6; % arbitrary delta t
mu = 398600;
options = odeset('RelTol',1e-5); %ode45 accuracy options
[T,Z] = ode45('OrbEq',[ti tf],[ro vo],options); % time range of ti to
tf, and inital conditions of ro,vo

R = 6378; % radius of Earth. Used for plot
%pretty image
A=imread('earth2.jpg');
[mytext,map]=rgb2ind(A,256); %usable data for matlab
mytext=flipud(mytext);
hold on;
% Create the surface.
[xx,yy,zz] = sphere(100); % generates sphere data
colormap(map)
surface(R*xx,R*yy,R*zz, 'CData',mytext,'FaceColor','texturemap','FaceLighting','phong')
%set the colormap.
hold on;
plot3(Z(:,1),Z(:,2),Z(:,3),'LineWidth',2,'Color','k') % plots the
x,y,z vectors from ode45
axis([-1.5e04 1.5e04 -1.5e04 1.5e04 -1.5e04 1.5e04]) % axis size to
keep sphere spherical
line([2*R 0 0],[0 0 0],[0 0 0],'LineStyle','--');text(2*R, -R, 0, 'To
Vernal Equinox'); %axis and labels
line([0 0 0],[0 2*R 0],[0 0 0],'LineStyle','--');text(0, 2*R,
0, 'Y');
line([0 0 0],[0 0 0],[0 0 2*R],'LineStyle','--');text(0, 0,
2*R, 'Zenith');
view( [2 1.5 0.8]) % specific view point
grid
r = [Z(:,1) Z(:,2) Z(:,3)]; % concatenating solution back into state
vectors
v = [Z(:,4) Z(:,5) Z(:,6)];
for i = 1:length(T) % normalizes velocity vector for each time step
speed(i) = norm(v(i,:));
end
[V_p,j_p] = max(speed); % velocity at apogee will be at a minimum and
occurs at the (j_a)th time step
[V_a,j_a] = min(speed); % same for perigee, but at a maximum
r_vec_p = r(j_p,:); % state vector at j_pth time step
v_vec_p = v(j_p,:);
r_vec_a = r(j_a,:);
v_vec_a = v(j_a,:);
```

```

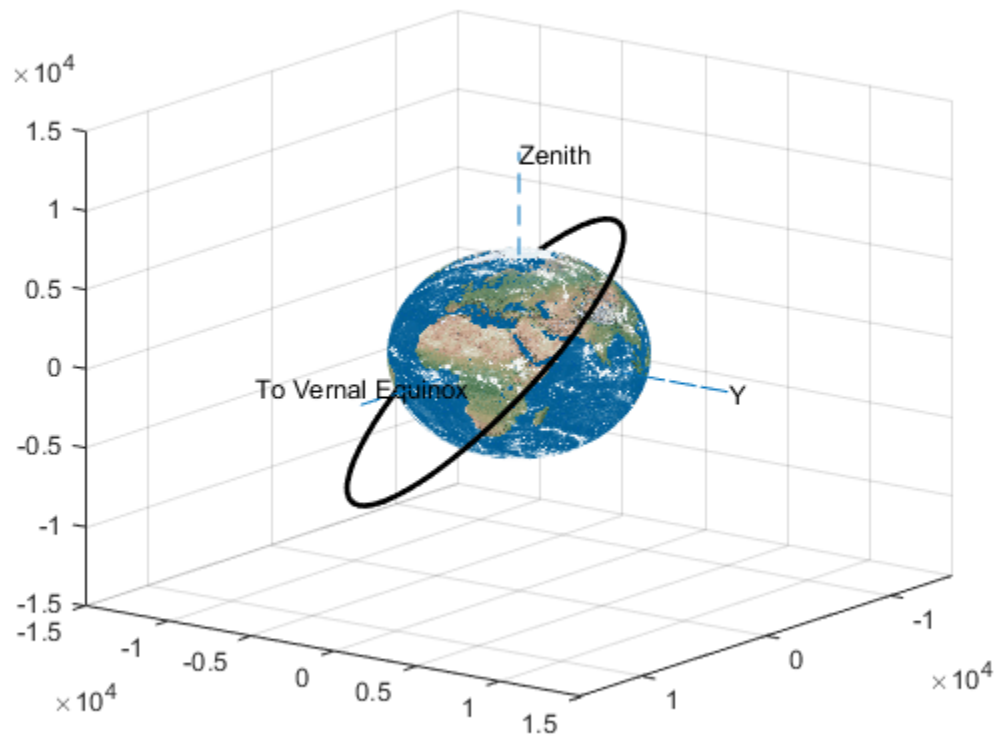
fprintf('Position at Apogee: [%0.f, %0.f, %0.f] km \n', r_vec_a)
fprintf('Distance at Apogee %0.2f km \n', norm(r_vec_a))
fprintf('Velocity at Apogee: [%0.2f, %0.2f, %0.f] km/s \n', v_vec_a)
fprintf('Speed at Apogee: %0.2f km/s \n', norm(v_vec_a))

fprintf(' \n')
fprintf('Position at Perigee: [%0.f, %0.f, %0.f] km \n', r_vec_p)
fprintf('Distance at Perigee %0.2f km \n', norm(r_vec_p))
fprintf('Velocity at Perigee: [%0.2f, %0.2f, %0.f] km/s \n', v_vec_p)
fprintf('Speed at Perigee: %0.2f km/s \n', norm(v_vec_p))

Position at Apogee: [-7386, -12746, -6421] km
Distance at Apogee 16069.32 km
Velocity at Apogee: [2.82, -0.29, -3] km/s
Speed at Apogee: 3.86 km/s

Position at Perigee: [3207, 5459, 2714] km
Distance at Perigee 6888.49 km
Velocity at Perigee: [-6.53, 0.78, 6] km/s
Speed at Perigee: 9.00 km/s

```



Published with MATLAB® R2017a