```matlab
% Author: Tatiana D. Luna

%QUESTION 1
%converting orbital parameters to ECI cartesian CS with 313
G = 6.6742*10^(-20); %km^3/(kg*s^2) gravitational constant
m1 = 5.972*10^24; %kg mass of Earth
m2 = 0; %negligible compared to Earth
mu = G*(m1+m2); %km^3/s^2 gravitational parameter

a = 6788.70030; %km
e =  0.0017144; %enorm
i = 51.30505*pi/180; %inclination
RAAN= 284.05125*pi/180; %right ascension of the ascending node
w = 128.76998*pi/180; %w is argument of perigee
theta = 39.75224*pi/180 ; %true anomaly

h = sqrt(a*mu*(1-e^2));
r = (h^2)/(mu)*(1/(1+e*cos(theta)));

%finding V vector
R1_i = [1 0 0; 0 cos(i) sin(i); 0 -sin(i) cos(i)];
R3_w = [cos(w) sin(w) 0; -sin(w) cos(w) 0; 0 0 1];
R3_RAAN = [cos(RAAN) sin(RAAN) 0; -sin(RAAN) cos(RAAN) 0; 0 0 1];
% R1_i = [1 0 0; 0 cosd(i) sind(i); 0 -sind(i) cosd(i)];
% R3_w = [cosd(w) sind(w) 0; -sind(w) cosd(w) 0; 0 0 1];
% R3_RAAN = [cosd(RAAN) sind(RAAN) 0; -sind(RAAN) cosd(RAAN) 0; 0 0 1];

R = R3_w*R1_i*R3_RAAN;

x = R(1,1:3);
y = R(2,1:3);
z = R(3,1:3);

e_orb = e*x;
h_orb = h*z;

rnorm = ((h^2)/mu)*(1/(1+e*cos(theta)));
r = (rnorm*cos(theta)*x + rnorm*sin(theta)*y)';
u_r = r/norm(r);
u_per = (cross(z, u_r)/norm(cross(z, u_r)))';
v = (mu/hnorm)*enorm*sin(true_anomaly)*u_r +
(mu/hnorm)*(1+enorm*cos(true_anomaly))*u_per;

%QUESTION 2
hvector = cross(rv, v);
evector = cross(v, hvector)/mu - (rv/r); %[-] eccentricity vector
rp = (h^2/mu)*(1/(1+e)); %km perigee
ra = (h^2/mu)*(1/(1-e)); %km apogee

%numerical integration of eqq. of motion
until= 15*60*60; %integrating for 15 hrs
TSPAN = [0 until]; %duration of integration in seconds
OPTIONS = odeset('Maxstep', 10);
Y0 = [rv'; v'];
[TOUT,YOUT] = ode45(@relacc_with_j2,TSPAN,Y0,OPTIONS);
```

```
%plotting orbit from ODE45
figure(1)
plot3(YOUT(:,1),YOUT(:,2),YOUT(:,3),'r','linewidth',2)
grid on
hold on
xlabel('x (km)'); ylabel('y (km)'); zlabel('z (km)');

load('topo.mat', 'topo', 'topomap1');
[X,Y,Z] = sphere;
props.FaceColor= 'texture';
props.EdgeColor = 'none';
props.FaceLighting = 'phong';
props.CData = topo;
s=surface(6378*X,6378*Y,6378*Z,props);

%QUESTION 3
orbital = state_to_orbital(YOUT);
fileID = fopen('Orbitalparameters.txt', 'w');

fprintf(fileID,'Orbital parameters \r\n      h          e    Inclanation
RAAN    arg_per  theta \r\n');
for n=1:21601
fprintf(fileID,'%f %f %f %f %f %f \r\n', orbital(n,1), orbital(n,2), ...
    orbital(n,3),orbital(n,4), orbital(n,5), orbital(n,6));
end
fclose(fileID);
open('Orbitalparameters.txt');

%QUESTION 4
display('Press enter to continue');
pause
clf
plot(TOUT,YOUT)
grid on
hold on
xlabel('time [s]'); ylabel('EA [deg]')
legend('\Omega', 'i', '\omega')
pause
display('Press enter to continue')

%QUESTION 5
date=[2016,02,20,12,25,0];
julianDATE = juliandate(date);
gst=gstime(julianDATE);
ang_vE = 72.9217*10^-6; %earths angular velocity, rad/s

k= size(YOUT);
for z=1:k(1,1)
    lat(z,1)= asind(YOUT(z,3)/norm(YOUT(z,1:3)));
    lon(k,1) = atan2(YOUT(z,2),YOUT(z,1))+ang_vE*z;
end

clf
plot(lat);
```

```
ylabel('Latitude');
display('Press enter to continue')
pause
clf


%QUESTION 6 and 7
Kepler = orbital(1,:);
GMSTo = gst;
Tf= until;

r2d = 180/pi;  %Radians to degrees conversion
w_earth  = 7.2921158553e-5; %rad/s  %Rotation rate of Earth

    h = figure;
    dT  = 60;
    s   = '.';
    mu  = 398600.4415;     %km^3/s^2  Gravitational constant of Earth

a   = Kepler(1);
ecc = Kepler(2);
inc = Kepler(3);
O   = Kepler(4);
w   = Kepler(5);
nuo = Kepler(6);

n = (mu/a^3)^.5;  %Mean motion of satellite

E  = atan2(sin(nuo)*(1-ecc^2)^.5,ecc+cos(nuo));  %Eccentric anomaly
MA = E-ecc*sin(E); %Initial Mean anomaly

time = [0:dT:Tf]; %time vector

bp(1)    = 0;
bp(2)    = length(time);
k        = 2;
Lat_rad  = zeros(1,length(time));
Long_rad = zeros(1,length(time));

for j = 1:length(time)
    GMST = zeroTo360(GMSTo + w_earth*time(j),1);  %GMST in radians

    M  = zeroTo360(MA + n*time(j),1);  %Mean anomaly in rad
    nu = nuFromM(M,ecc,10^-12);        %True anomaly in rad

    [ECI,Veci] = randv(a,ecc,inc,O,w,nu);
    clear veci
    ECEF = eci2ecef(ECI,GMST);

r_delta = norm(ECEF(1:2));

sinA = ECEF(2)/r_delta;
cosA = ECEF(1)/r_delta;
```

```
Long = atan2(sinA,cosA);

if Long < -pi
    Long = Long + 2*pi;
end

Lat = asin(ECEF(3)/norm(ECEF));
    Lat_rad(j) = Lat;
    Long_rad(j) =Long;
end

Lat              = Lat_rad.*r2d;
Long             = Long_rad.*r2d;

x = load('Coastline.dat'); %loads earth image
plot(x(:,1)+360,x(:,2),'black', 'linewidth',1) %plots it
hold on

%QUESTION 8
plot(360-82.3250, 29.6520, 'p','linewidth',2) %plotting Gainesville
plot(Long+360, Lat, s)
xlabel('East Longitude (\circ)')
ylabel('Latitude (\circ)')
plot(x(1:30816,1),x(1:30816,2),'black', 'linewidth',1)
clear x
plot(Long+180, Lat, s)
axis([0,360,-90,90])

%QUESTION 9
% GvilleLat = 29.6520;
% Glong = 180-82.325;
% [c index] = min((Lat-GvilleLat))
% closestValues = Lat(index)
for g=1:901
    dist(g,1)=sqrt((Lat(1,g)-29.65)^2+(Long(1,g)-82.32)^2);
end
[distance, time]=find(min(dist));
adjust=mod(10*time,60);
B=(10*time-adjust)/60;
q=mod(B,60);
b=(B-q)/60; %hours it has taken
sec=adjust; min=s; hr=b;
mmiin=[hr min sec];
fprintf('It is closest to Gainesville at 10 seconds\n');

%QUESTION 10

rfin = 10000;
[dV,T,a_tx]=Hohmann(r,rfin);
deltaV= abs(norm(v)-norm(dV))




%QUESTION 11
```

```
fprintf('Between two circular orbits the most efficient manuever is the
Hohmann Transfer\n');
fprintf('Therefore the bi-elliptic is not the most efficient \n');

%EXTRA CREDIT
fprintf('Extra credit question: drag \n');
```