Problem 1

```
clear; close all; clc; format long

mu_e = 1; % Mu earth in DU^3/TU^2

r = [0.33; -0.40; -0.71]'; % in DU
v = [-0.20; -0.20; -0.25]'; % in DU/TU

r_mag = sqrt(r*r'); % magnitude of r in DU
v_mag = sqrt(v*v'); % magnitude of v in DU/TU


H = cross(r,v); % Specific Angular momentum in DU^2/TU
H_mag = sqrt(H*H'); % magnitude of v in DU^2/TU

energy = 0.5*v_mag^2 - mu_e/r_mag; % total spcific energy of the Orbit
                                   % in DU^2/TU^2

a = -mu_e/(2*energy); % Semi-Major Axis in AU

e = cross(v,H)/mu_e - r/r_mag; % e_hat unit less

e_mag = sqrt(e*e'); % Eccentricity unit less

k = [0 0 1]; % unit vector of the earths spin axis

cos_i = k*H'/H_mag;

i = acos(cos_i); % inclination in rad

kcrossH = cross(k,H);
kcrossH_mag = sqrt(kcrossH*kcrossH');

n_hat = kcrossH/kcrossH_mag; % n_hat unit less

ni = n_hat(1);
nj = n_hat(2);

Omega = atan2(nj,ni); % Right Ascension of the ascending node in rad

omega = acos(n_hat*e'/e_mag); % Argument of Perigee in rad

nu = acos(r*e'/(r_mag*e_mag)); % True Anomoly in rad

disp('The six orbital elements are,');
```

```
disp('Semi-major axis, a, (DU):');
disp(a);
disp('Eccentricity, e, (Unit Less):')
disp(e_mag);
disp('Inclination, i, (deg):')
disp(rad2deg(i));
disp('Right Ascension of the ascending node, \Omega, (deg):')
disp(rad2deg(Omega));
disp('Argument of Perigee, \omega, (deg):')
disp(rad2deg(omega));
disp('True Anomoly, \nu, (deg):')
disp(rad2deg(nu));
```

```
The six orbital elements are,
Semi-major axis, a, (DU):
   0.468980583351249

Eccentricity, e, (Unit Less):
   0.918324129249912

Inclination, i, (deg):
     122.5884862476130

Right Ascension of the ascending node, \Omega, (deg):
    -169.4034602872141 or (190.5965397127859)

Argument of Perigee, \omega, (deg):
  77.110974152316331

True Anomoly, \nu, (deg):
     176.3137003005811
```

Problem 2

```
clear

mu_s = 1; % Mu sun in AU^3/TU^2
r_e = 1; % Radius of Earth's orbit in AU
r_j = 5.203; % Radius of Jupiter's orbit in AU

a1 = r_e; % circular Earth orbit
a2 = r_j; % circular Jupiter orbit

Delta_v_1 = sqrt(mu_s/a1)*( sqrt(2*a2/(a1+a2)) - 1 ); % to enter transfer orbit
Delta_v_2 = sqrt(mu_s/a2)*( 1 - sqrt(2*a1/(a1+a2)) ); % to circularize the orbit at Jupiter

TOF = pi*sqrt((a1+a2)^3 / (8*mu_s));

n_e = sqrt(mu_s/r_e^3); % mean motion at Earth's orbit
n_j = sqrt(mu_s/r_j^3); % mean motion at Jupiter's orbit

phi_l = TOF*n_j; % required phase angle in rad
```

```
phi_p = pi - phi_l;

phi_dot = n_j - n_e;
phi_0 = pi;

t_burn = (phi_p - phi_0)/phi_dot ; % Wait time is TU

disp('The Hohmann Orbit Transfer parameters are,')
disp('Detal V required enter to transfer orbit, \delta v_1, in (AU/TU):')
disp(Delta_v_1)
disp('Detal V required circularize the orbit at Jupiter, \delta v_2, in (AU/TU):')
disp(Delta_v_2)
disp('Time to reach Jupiter, TOF, (Days):')
disp(TOF*58.132821)
disp('Required phase angle for the spacecraft before maneuver start, \phi_p, (deg)')
disp(rad2deg(phi_p))
disp('wait time to begin the journey, t_burn, (Days):')
disp(t_burn*58.132821)
```

```
The Hohmann Orbit Transfer parameters are,
Detal V required enter to transfer orbit, \delta v_1, in (AU/TU):
   0.295212479386305

Detal V required circularize the orbit at Jupiter, \delta v_2, in (AU/TU):
   0.189466859404141

Time to reach Jupiter, TOF, (Days):
     997.5368532449240

Required phase angle for the spacecraft before maneuver start, \phi_p, (deg)
  97.158215686119604

wait time to begin the journey, t_burn, (Days):
  91.785868958414511
```

Problem 3

```
clear

alt_perigee = 375; % Altitude of the perigee in km
alt_apogee = 35786; % Altitude of the apogee in km
oneDU = 6378.145; % mean equatorial radius in km
oneTU = 806.8118744; % one TU in sec

r_ap = 1 + alt_apogee/oneDU; % r at apogee in DU
r_pr = 1 + alt_perigee/oneDU; % r at perigee in DU
r_e = 1; % mean equatorial radius in DU
mu_e = 1; % mu earth in DU^3/TU^2

a = 0.5*(r_ap+r_pr); % Semi-major axis in DU
e = (r_ap - a)/a; % eccentricity
b = a*sqrt(1-e^2); % Semi-minor axis in DU
```

```
n = sqrt(mu_e/a^3); % mean motion 1/TU

y_sh = r_e; % y cordinate on the ellipse where the shadow ends
x_sh = sqrt( a^2 * (1 - (y_sh^2/b^2)) ) - a*e; % x cordinate on the ellipse
                                               % where the shadow ends
                                               % computed from the equation
                                               % of ellipse


nu_sh = atan2(y_sh,x_sh); % true anomoly where the shadow ends on GTO
E_sh = 2 * atan( sqrt((1-e)/(1+e)) * tan(nu_sh/2)); % Eccentric anomoly
                                                    % where the shadow ends
                                                    % on the GTO


M_sh = E_sh - e*sin(E_sh); % Mean anomoly where the shadow ends on GTO

t_sh = M_sh/n; % time spent in shadow in TU

T_GTO = 2*pi/n; % Time period of GTO in TU

t_bw_sh_and_burn = T_GTO/2 - t_sh; % Tine spent between end of shadow
                                   % and the final burn, in GTO

nu_sh_gto = pi - asin(r_e/r_ap); % true anomoly where the shadow begin
                                 % in the geosynchronous orbit (circular)

disp('Charatersitics of satellite in GTO in Earth eclispse')
disp('Time spent in sunlight in GTO, (TU):')
disp(t_bw_sh_and_burn)
disp('Time spent in sunlight in GTO, (sec):')
disp(t_bw_sh_and_burn*oneTU)
disp('True anomoly in GTO where shadow ends, \nu_sh, (deg):')
disp(rad2deg(nu_sh))
disp('True anomoly in geosynchronous orbit where shadow begins, \nu_sh, (deg):')
disp(rad2deg(nu_sh_gto))
```

```
Charatersitics of satellite in GTO in Earth eclispse
Time spent in sunlight in GTO, (TU):
  22.735528783032393

Time spent in sunlight in GTO, (sec):
     1.834329459291352e+04

True anomoly in GTO where shadow ends, \nu_sh, (deg):
  52.249129818336186

True anomoly in geosynchronous orbit where shadow begins, \nu_sh, (deg):
     171.2995025780532
```

Problem 4

```
clear;

J2 = 0.001082; % Oblateness of earth unit less
r_e = 1; % radius of Earth in DU
i = deg2rad(52); % inclination in rad
omega_0 = deg2rad(117.5); % inital argument of perigee in rad
oneDU = 6378.145; % mean equatorial radius in km
oneTU = 806.8118744; % one TU in sec

sec_in_year = 365.25*24*3600;
TU_in_year = sec_in_year/oneTU;

e = 0.02; % eccentricity
r_p = 7098/oneDU; % r at perigee in DU
mu_e = 1; % mu earth in DU^3/TU^2
a = r_p/(1-e); % Semi-major axis in DU
n = sqrt(1/a^3); % mean motion 1/TU
p = a*(1-e^2); % semi-latus-rectum in DU

omega_dot = - (2.5*sin(i)^2 - 2) * (3*n*J2*r_e^2) / (2*r_p^2); % in rad/TU

omega_thr = deg2rad(122.5); % in rad

t_to_man = (omega_thr-omega_0)/omega_dot; % in TU

delta_v_for_oneTU = 2*e*sqrt(mu_e/p)*sin(omega_dot/2); % required delta v per TU in DU/TU
delta_v_per_year = delta_v_for_oneTU*TU_in_year; % required delta v per year in DU/TU

disp('Correction for the Satellite')
disp('Time to first correction, (days)')
disp(t_to_man*oneTU/3600/24)
disp('Delta V requirements per year, \Delta v/yr, (DU/TU)')
disp(delta_v_per_year)
```

```
Correction for the Satellite
Time to first correction, (days)
   1.681148674299288

Delta V requirements per year, \Delta v/yr, (DU/TU)
   0.355910316053011
```

Published with MATLAB® R2019a