AERO-423, Spring 2024, Homework #3
(Due date: 23:59 hours, Thursday, March 21, 2024)

---

1. Show all the work and justify your answers.
2. Upload your submission to the CANVAS, in time.
3. This is a long homework. Start right now!

1-a. (8 points) An astronomer observes the following position $\boldsymbol{r}$ and velocity $\boldsymbol{v}$ vectors of the ISS in the inertial frame.

$$\boldsymbol{r} = \begin{Bmatrix} -10,063.829 \\ -473.07 \\ -12,487.599 \end{Bmatrix} \text{ (km)} \qquad \text{and} \qquad \boldsymbol{v} = \begin{Bmatrix} -0.359 \\ -4.950 \\ 0.475 \end{Bmatrix} \text{ (km/s)}$$

Derive the Keplerian ISS orbital elements: a, e, i, Ω, ω and the true anomaly, φ. Use $\mu = 398,600$ km/s^2.

Solution:

$$\begin{aligned} a &= 16054.449 \, \text{km} \\ e &= 0.000651 \\ i &= 0.9013 \, \text{rad} \quad \text{or} \quad 51.64^\circ \\ \Omega &= 4.564 \, \text{rad} \quad \text{or} \quad 261.54^\circ \\ \omega &= 4.311 \, \text{rad} \quad \text{or} \quad 250.75^\circ \\ \varphi &= 0.458 \, \text{rad} \quad \text{or} \quad 26.244^\circ \end{aligned}$$

1-b. (8 points) Convert the following orbital elements,

$$\begin{Bmatrix} h \\ i \\ \Omega \\ e \\ \omega \\ \varphi \end{Bmatrix} = \begin{Bmatrix} 80,000 \, \text{km}^2/\text{s} \\ 51.2^\circ \\ 250^\circ \\ 0.00083 \\ 240^\circ \\ 15^\circ \end{Bmatrix}$$

into Cartesian coordinates, position $\boldsymbol{r}$ (km) and velocity $\boldsymbol{v}$ (km/s).

Solution:

$$\boldsymbol{r} = \begin{Bmatrix} -7704.494 \\ -7223.013 \\ -12077.144 \end{Bmatrix} \text{ (km)} \qquad \text{and} \qquad \boldsymbol{v} = \begin{Bmatrix} -2.4078 \\ -4.249 \\ -1.0066 \end{Bmatrix} \text{ (km/s)}$$

2. Coding problem (14 pts). Two spacecrafts A and B are in a same circular orbit of radius $R = 13,600$ km as shown in the Figure 2. Spacecraft B is initially at β angle ahead of A. Perform a lower rendezvous maneuver to rendezvous with B by $k = 5$ revolutions.

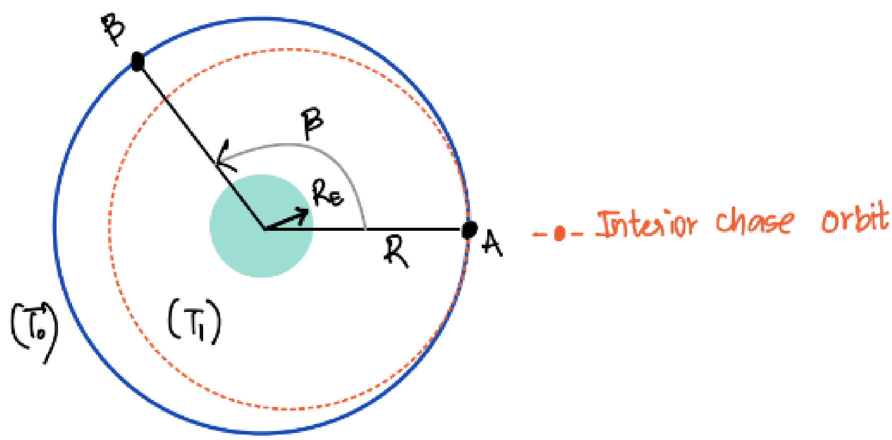


Figure 1: Rendezvous geometry for problem #2

Write a code that plot the total Δv as a function of the angle $\beta \in [10, 350]$ deg and the altitude (in km) of the perigee of the transfer orbit

Solution:

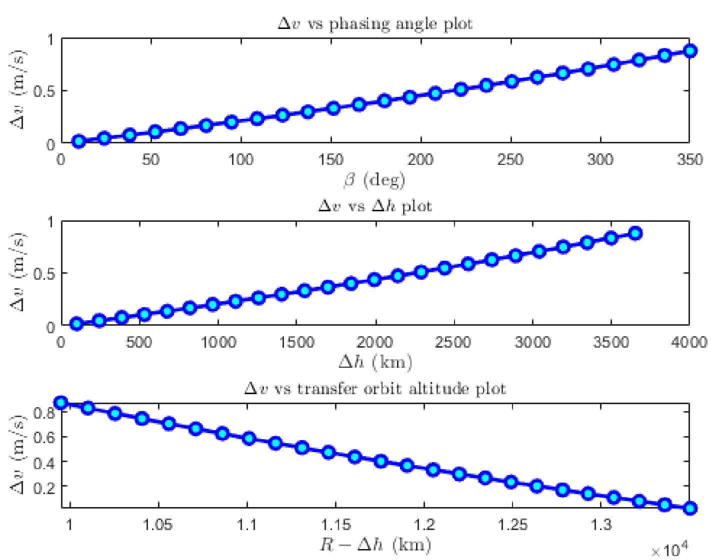


Figure 2: Solution for problem 2: Rendezvous plots

$$
\begin{aligned}
T_1 &= T_0 - \frac{\beta}{k n_0} \\
\Delta h &= 2(R - a) \\
\Delta v_{\text{tot}} &= 2\Delta v = 2\sqrt{\frac{\mu}{R}}\left(1 - \sqrt{\frac{2(R - \Delta h)}{2R - \Delta h}}\right)
\end{aligned}
$$

3. (15 pts) For the following two cases determine the required launch azimuth for a satellite it is launched from Cape Canaveral (latitude = 28.5°N). Also answer the following for each of the cases (a) is direct launch to specified parking orbit feasible and why?, (b) if applicable, specify when to apply orbit plane corrections, (c) if applicable, specify the Δv impulse for the correction. Consider $\mu = 398600$ km^3/s^2. Angular momentum direction $\hat{\boldsymbol{h}} = \left\{-\sin i \, \sin\Omega, \quad \sin i \, \cos\Omega, \quad \cos i\right\}^{\mathrm{T}}$. The

3-a. (5 points) A satellite is to be launched into a circular orbit with a period of 100 minutes, inclination of 98.43°, $\Omega = 105°$.

Solution:

a. Yes, direct launch to parking orbit is feasible because

$$|l| \leq i \leq \pi - |l|$$

Launch azimuth, $\sin A = \frac{\cos i}{\cos l} = 189.6°$ suggesting a launch trajectory due south or retrograde launch which is rare in practice. This problem is not about checking the launch feasibility and is purely mathematical.

b. No orbit plane corrections necessary.

c. No Δv corrections are necessary as orbit plane changes are not occurring.

3-b. (10 points) A satellite is to be launched into a circular orbit with a period of 100 minutes, inclination of 28° and $\Omega = 105°$.

Solution:

a. No. Because parking orbit's inclination is less than the latitude, direct launch is not possible. The best option is to launch to East with azimuth at launch of $A = 90°$.

b. The orbit plane correction can be applied at descending node or at ascending node. The descending node is met at $t = T/4 \approx 25$ minutes (or) 1500 s, while the descending node is met at $t = 3T/4 = 75$ minutes (or) 4500 s, where $T = 100$ minutes, is the orbital period that is given.
Here is a YouTube video on launch azimuth and orbit inclination: Launch site latitude, Orbit inclination.

c. The plane change impulse is,

$$
\begin{aligned}
\Delta v &= 2v\sin(\frac{\Delta i}{2}) \\
\Delta v &= 2 \times 7.4735 \times \sin(0.5 \times \pi/180) \approx \boxed{0.0652 \text{ km/s.}}
\end{aligned}
$$

where $\Delta i = 28.5° - 28° = 0.5°$, and v is computed from T using the equations

$$T = 2\pi\sqrt{\frac{R^3}{\mu}} \quad \text{(we get } R = 7136.63 \text{ km)}$$

$$v = \sqrt{\frac{\mu}{R}} \quad \text{(we get } v = 7.4735 \text{ km/s)}$$

4. Coding problem (15 pts) A spacecraft is in a 300 km circular Earth orbit. Use $\mu = 398,600.4415$ km^3/s^2 and $R_E = 6,378.135$ km. Calculate the total Δv required for the bi-elliptic transfer to a 3,000 km altitude co-planar circular orbits. Use bisection approach to find the root (unknown radius "r_b").

Solution:

- $r_A = R_E + 300 = 6,678.14$ km
- $r_C = R_E + 3,000 = 9,378.14$ km
- From numerical methods, the Bisection method here, $r_B = 7894.772$ km. Note that the r_B is less than r_C which is not really practical but is only of mathematical importance. You will find that $\frac{d\Delta_v}{dr_b}$ is almost zero for any higher values of r_b. You have to identify this to get full points for this problem.
- $\Delta v_{\text{tot}} = |v_{A_f} - v_{A_i}| + |v_{B_f} - v_{B_i}| + |v_{C_f} - v_{C_i}| = \boxed{1.204 \text{ km/s}}$

5. (15 pts) Using the data provided in fig. 3, consider the mission Saturn to Mars and back to Saturn. Consider all planets in circular orbits (radius = semi-major axis) and at inclination $i = 0$. Compute the following,

1. The departure angle;
2. The total Δv
3. The waiting time in Mars for starting an Homann transfer back to Saturn.

Solution:

1. 1.755 radians or 100.59°

$$R_s = 1.433 \times 10^9 \,\text{km}, \quad R_m = 227.9 \times 10^6 \,\text{km}, \quad \mu_{\text{sun}} = 132.71 \times 10^9 \text{km}^3/\text{s}^2$$

$$a_{\text{tr}} = \frac{R_s + R_m}{2} = 830450000 \,\text{km}$$

$$t_{\text{tr}} = \pi\sqrt{\frac{a_{\text{tr}}^3}{\mu_{\text{sun}}}} = 2.0638 \times 10^8 \,\text{s}$$

Mars completes many revolutions during the transfer time from Saturn.

$$\theta_{SM}^{\text{dep}} = \pi - mod(n_m t_{\text{tr}}, 1) = 2.289 \,\text{rad} = 131.14°$$

2. 24.299 km/s

$$\Delta V_{\text{dep}} = \sqrt{\frac{\mu_{\text{sun}}}{R_s}}\left(1 - \sqrt{\frac{2R_m}{R_m + R_s}}\right) = 4.5821\,\text{km/s}$$

Spacecraft's heliocentric speed at departure is less than that of the Saturn.

The spacecraft is slowing down to transition to the transfer orbit toward Mars.

$$\Delta V_{\text{arr}} = \sqrt{\frac{\mu_{\text{sun}}}{R_m}}\left(1 - \sqrt{\frac{2R_m}{R_m + R_s}}\right) = 7.5678\,\text{km/s}$$

At arrival, spacecraft on the transfer ellipse is faster than the Mars.

$\Delta v_{\text{SM}} = \Delta V_{\text{dep}} + \Delta V_{\text{arr}} = 12.1498\,\text{km/s}$

$\Delta v_{\text{total}} = 2 \times \Delta v_{\text{SM}} \approx 24.299\,\text{km/s}$

3. Waiting time in Mars for starting Hohman Transfer back to Saturn is 323.51 days or 2.795×10^7 s.

$$t_{\text{wait}} = 2\frac{n_s t_{\text{tr}}}{n_m - n_s} = 2.795 \times 10^7 \text{secs}$$

where $n_s = 6.7156 \times 10^{-9}\ \text{s}^{-1}$.

Object	Radius (km)	Mass (kg)	Sidereal Rotation Period	Inclination of Equator to Orbit Plane	Semimajor Axis of Orbit (km)	Orbit Eccentricity	Inclination of Orbit to the Ecliptic Plane	Orbit Sidereal Period
Sun	696,000	1.989×10^{30}	25.38d	7.25°	—	—	—	—
Mercury	2440	330.2×10^{21}	58.65d	0.01°	57.91×10^6	0.2056	7.00°	87.97d
Venus	6052	4.869×10^{24}	243d*	177.4°	108.2×10^6	0.0067	3.39°	224.7d
Earth	6378	5.974×10^{24}	23.9345h	23.45°	149.6×10^6	0.0167	0.00°	365.256d
(Moon)	1737	73.48×10^{21}	27.32d	6.68°	384.4×10^3	0.0549	5.145°	27.322d
Mars	3396	641.9×10^{21}	24.62h	25.19°	227.9×10^6	0.0935	1.850°	1.881y
Jupiter	71,490	1.899×10^{27}	9.925h	3.13°	778.6×10^6	0.0489	1.304°	11.86y
Saturn	60,270	568.5×10^{24}	10.66h	26.73°	1.433×10^9	0.0565	2.485°	29.46y
Uranus	25,560	86.83×10^{24}	17.24h*	97.77°	2.872×10^9	0.0457	0.772°	84.01y
Neptune	24,760	102.4×10^{24}	16.11h	28.32°	4.495×10^9	0.0113	1.769°	164.8y
(Pluto)	1195	12.5×10^{21}	6.387d*	122.5°	5.870×10^9	0.2444	17.16°	247.7y

Figure 3: Solar system data (from Curtis)

MATLAB Code:

AERO 423 - HW3

Problem 1 - Orbital Elements

PART A

[h,i,OMEGA,e,omega,theta] from [r,v] specified in geocentric equatorial frame. Is ISS in retrograde orbit? Is the satellite flying towards or away from perigee?

Period of the orbit?

What are the position and velocity vectors of the satellite in the perifocal frame precisely 24 hours later.

```
r = [-10063.829, -473.07, -12487.599];
v = [-0.359, -4.950, 0.475];

[h,i,OMEGA,e,omega,phi] = cartesian2OrbitElements( r,  v);
```

```
v_r = 0.001434 > 0: satellite is flying away from perigee
Not in retrograde orbit
```

```
mu = 398600;

r_p = h^2 / mu * ( 1/(1+e) );
r_a = h^2 / mu * ( 1/(1-e) );
a = .5 * (r_p + r_a);
T = 2*pi /sqrt(mu) * a^(1.5);

fprintf("a = %f km,\n e = %f,\n i = %f deg,\n OMEGA = %f deg,\n ..." + ...
    "omega = %f deg,\n phi=%f\n deg", a,e,i*180/pi,OMEGA*180/pi,omega*180/pi,phi*180/pi)
```

```
a = 16054.449583 km,
 e = 0.000651,
 i = 51.639276 deg,
 OMEGA = 261.507519 deg,
 ...omega = 250.751280 deg,
 phi=26.244051
 deg
```

```
fprintf( 'Orbital period is: %f hr\n', T/(3600) );
```

```
Orbital period is: 5.623437 hr
```

PART B

[r, v] in perifocal and geocentric equatorial frames. MU = 398600.

```
h = 80e3;
e = 0.00083;
i = deg2rad(51.2);
OMEGA = deg2rad(250);
```

```
omega = deg2rad(240);
theta = deg2rad(15);
[r,v] = orbit2GeocentricCartesian(h,e,i,OMEGA,omega,theta);

fprintf('r = \r'); fprintf('%f\r', r);
```

```
r =
-7704.494787
7223.013956
-12077.144108
```

```
fprintf('v = \r'); fprintf('%f\r', v);
```

```
v =
-2.407807
-4.249037
-1.006619
```

Problem 2

Lecture: 423-07, pg. 6

Higher transfer orbit example problem in the lecture

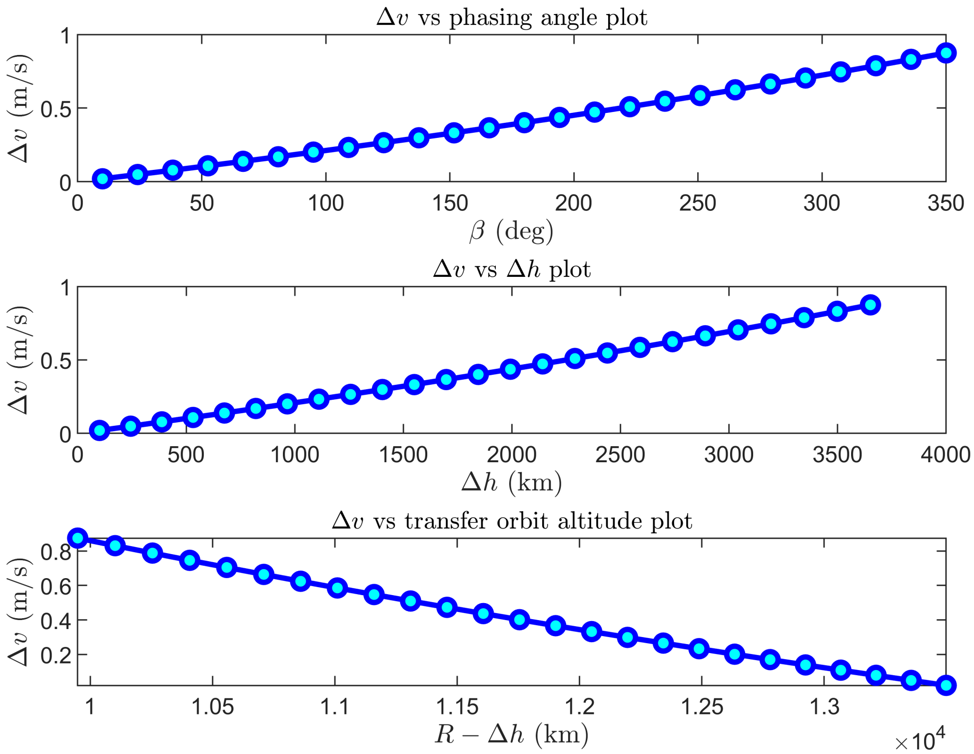
```
R = 7000;
k = 10;
beta = 270*pi/180;
mu = 398600;


T0 = 2*pi*sqrt(R^3/mu);
n0 = 2*pi/T0;

T1 = T0 + ( 2*pi - beta ) / ( k*n0 );

a = ( mu * ( T1/(2*pi) )^2 )^(1/3);

va = sqrt( 2*mu/R - mu/a );
vb = sqrt( mu/R );
dV = 2*( va-vb )
```

```
dV = 0.1227
```

Lower transfer orbit HW #3 question:

```
R = 13600;
k = 5;
% beta = 270*pi/180;
mu = 398600;

T0 = 2*pi*sqrt(R^3/mu);
n0 = 2*pi/T0;
```

```
samples = 25;
beta = linspace(10, 350, samples);
dV = zeros(samples, 1);
dH = dV;

for ii = 1:samples
    T1 = T0 - deg2rad( beta(ii) ) / ( k*n0 );
    a = ( mu * ( T1/(2*pi) )^2 )^(1/3);
    dH(ii) = 2*(R - a);

    va = sqrt( mu/R * (2 * (R - dH(ii) ) / (2*R-dH(ii)) ) );
%     va = sqrt( 2*mu/R - mu/a );
    vb = sqrt( mu/R );
    dV(ii) = 2* ( vb-va );
end

figure;
subplot(3,1,1)
plot(beta, dV, '-ob','LineWidth',2, 'MarkerFaceColor','c');
xlabel("$\beta$ (deg)", "Interpreter","latex", "FontSize",10);
ylabel("$\Delta v$ (m/s)","Interpreter","latex", "FontSize",10);
title("$\Delta v$ vs phasing angle plot" , "Interpreter","latex", "FontSize",10);

subplot(3,1,2)
plot(dH, dV, '-ob','LineWidth',2, 'MarkerFaceColor','c');
xlabel("$\Delta h$ (km)", "Interpreter","latex", "FontSize",10);
ylabel("$\Delta v$ (m/s)","Interpreter","latex", "FontSize",10);
title("$\Delta v$ vs $\Delta h$ plot" , "Interpreter","latex", "FontSize",10);

subplot(3,1,3)
plot(R-dH, dV, '-ob','LineWidth',2, 'MarkerFaceColor','c');
xlabel("$R-\Delta h$ (km)", "Interpreter","latex", "FontSize",10);
ylabel("$\Delta v$ (m/s)","Interpreter","latex", "FontSize",10);
title("$\Delta v$ vs transfer orbit altitude plot" , "Interpreter","latex", "FontSize",10);

axis tight
```



Problem 3a

```
inclination = 98.43;
latitude    = 28.5;
az = launchAzimuth( inclination, latitude )
```

```
az = -9.6028
```

```

if az < 0
    az = 180 - az;
end

fprintf("Launch Azimuth A = %f\n", az);
```

```
Launch Azimuth A = 189.602761
```

Problem 3b

```
    T = 100*60; % orbit period in seconds
    mu = 398600;

    % descending node
    t_des = T/4;
```

```
    % ascending node
    t_asc = 3*T/4;

    fprintf("Descending node at t_des = %f and ascending node at t_asc = %f\n", t_des, t_asc);
```

```
Descending node at t_des = 1500.000000 and ascending node at t_asc = 4500.000000
```

```
    R = ( ( T / (2*pi) )^2 * mu )^(1/3)
```

```
R = 7.1366e+03
```

```
    v = sqrt( mu/R )
```

```
v = 7.4735
```

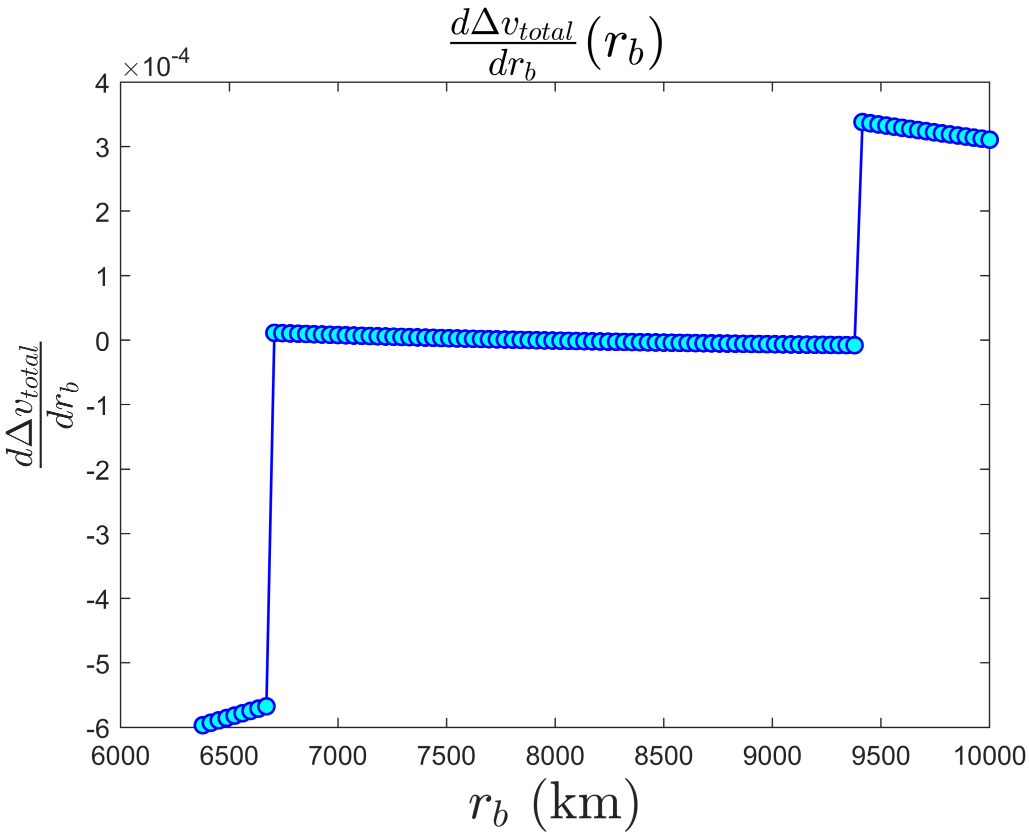
```
    % Delta v
    delta_i = 28.5 - 28;
    dV = 2*v*sind(delta_i/2)
```

```
dV = 0.0652
```

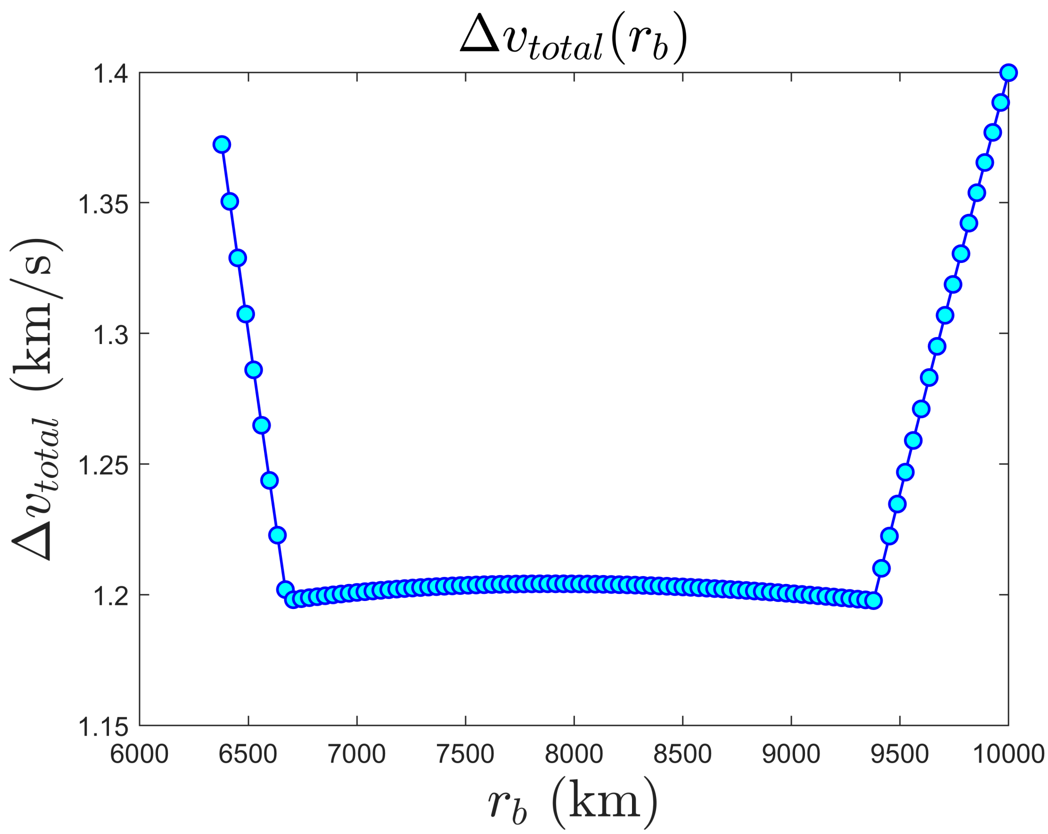
Problem 4

```
% Bisection method

% bounds

a = 6378;
b = 10000;

syms r_b real
dv_total_func = dV_total(r_b);   % look out for delta_v_total function below

% symbolic differentiation of delta_v_total wrt rb
DV_total = diff(dv_total_func, r_b);
rb_eqn = DV_total == 0;

rb_root = vpasolve(rb_eqn, r_b); % nonlinear root solver - my reference for bisection


% Plotting the obtained delta_v functions for find their nature
rb = linspace(a, b, 100);
dv_drb = zeros(100,1);
dv_total_func_vals = zeros(100,1);

for ii = 1:100
    dv_drb(ii) = subs(DV_total, r_b, rb(ii));
    dv_total_func_vals(ii) = subs(dv_total_func, r_b, rb(ii));
end
```

```
figure
plot(rb, dv_drb, '-ob', 'MarkerFaceColor','c', 'LineWidth', 1);
ylabel('$\frac{d \Delta v_{total}}{dr_b}$','Interpreter', 'Latex', 'FontSize',20);
xlabel('$r_b$ (km)','Interpreter', 'latex', 'FontSize',20);
title('$\frac{d \Delta v_{total}}{dr_b} (r_b)$','Interpreter', 'latex', 'FontSize',20);
```




```
figure
plot(rb, dv_total_func_vals, '-ob','MarkerFaceColor','c', 'LineWidth',1);
ylabel('$\Delta v_{total}$ (km/s)','Interpreter', 'Latex', 'FontSize',20);
xlabel('$r_b$ (km)','Interpreter', 'latex', 'FontSize',20);
title('$\Delta v_{total}(r_b)$','Interpreter', 'latex', 'FontSize',20);
```




```
% Bisection method initiation
iter = 0; tol = 1e-6; max_iter = 1000;

while (b - a) / 2 > tol && iter < max_iter

    rb = (a + b) / 2;
    dv_total_val = subs(DV_total, r_b, rb);

    if dv_total_val == 0
       break;
    elseif dv_total_val < 0
       b = rb;
    else
       a = rb;
    end
       iter = iter + 1;
end

val = (a + b) / 2;  % root for r_b
fprintf('Root for r_b is found after %d iterations: %f\n', iter, val);
```

```
Root for r_b is found after 31 iterations: 7894.772060
```

```
fprintf('Delta v total = %f\n', dV_total(val));
```

```
Delta v total = 1.204150
```

Problem 5

```
mu_sun = 132.71 * 1e9;
R_saturn = 1.433 * 1e9;
R_mars = 227.9 * 1e6;

T_mars = 2*pi*sqrt(R_mars^3/mu_sun);
T_saturn = 2*pi*sqrt(R_saturn^3/mu_sun);
n_mars = 2*pi / T_mars;
n_saturn = 2*pi / T_saturn;

% semi-major axis
a_transfer = (R_saturn + R_mars)/2;

t_transfer = pi * sqrt(a_transfer^3 / mu_sun);

% Part 1: Departure angle - Multirevolution problem
% Mars completes n revolutions in the transfer time
n_revs = n_mars * t_transfer;

theta_depature = pi - mod(n_revs, 1);
fprintf("At departure Saturn to Mars must be %f (rad) or %f deg\n", ...
    theta_depature, theta_depature*180/pi);
```

```
At departure Saturn to Mars must be 2.288967 (rad) or 131.148169 deg
```

```
% Part 2: Delta V
V_saturn = sqrt( mu_sun / R_saturn );
V_departure = sqrt(2*mu_sun)*sqrt(R_mars / (R_saturn * (R_mars + R_saturn) ));

% Required Delta_v at departure
dV_departure =  V_saturn - V_departure % saturn is faster
```

```
dV_departure = 4.5821
```

```
% another approach
% dV_departure = sqrt(mu_sun / R_saturn) * ( sqrt(2*R_mars/(R_mars + R_saturn) ) - 1 )

V_mars = sqrt( mu_sun / R_mars );
V_arrival = sqrt(2*mu_sun)*sqrt( R_saturn / ( R_mars * (R_mars + R_saturn) ) )
```

```
V_arrival = 31.6990
```

```
dV_arrival = V_arrival - V_mars ;

% dV_arrival = sqrt(mu_sun / R_mars) * ( sqrt(2*R_saturn/(R_mars + R_saturn) ) - 1 )
```

```
fprintf('Round trip delta v is %f km/s\n', 2*(dV_departure + dV_arrival));
```

```
Round trip delta v is 24.299754 km/s
```

```
% Part 3: Waiting time in Mars for starting Hohman Transfer back to Saturn

t_wait = 2 * ( (2*pi - n_saturn*t_transfer) / (n_mars - n_saturn) );
fprintf("Waiting time in Mars before transfering back to Saturn is %f years or %f days or %d s'
     t_wait/(24*60*60*365), t_wait/(24*60*60), t_wait);
```

```
Waiting time in Mars before transfering back to Saturn is 3.131804 years or 1143.108637 days or 9.876459e+07 s
```

```
pf = pi - n_saturn*t_transfer;

t_wait = -2*pf / (n_mars - n_saturn) + 2*pi*1 / (n_mars - n_saturn)
```

```
t_wait = 2.7951e+07
```

```
t_wait = t_wait / (24*3600)
```

```
t_wait = 323.5104
```

Functions

```
function dV_t = dV_total(rb)

% Problem 4: Bielliptic transfer - delta_v as a function of rb
   mu = 398600.44;
   Re = 6378.135;
   ra = Re + 300;
   rc = Re + 3000;

   va = sqrt( mu/ra );
   vc = sqrt( mu/rc );

   va1 = sqrt( 2*mu*rb / ( ra*(ra+rb)) );
   vb1 = sqrt (2*mu*ra / (rb*(ra+rb)) );

   vb2 = sqrt( 2*mu*rc / (rb*(rb+rc)) );
   vc2 = sqrt( 2*mu*rb / (rc*(rb+rc)) );

   dVa = va1 - va;
   dVb = vb2 - vb1;
   dVc = vc2 - vc;

   dV_t = abs(dVa) + abs(dVb) + abs(dVc);

end
```

```
function A = launchAzimuth(inclination, latitude)
    % all in degrees

    A = asin( cos(inclination * pi/180) / cos(latitude * pi/180) );
    A = A * 180/pi;

end
```

```
function [h_mag,i,OMEGA,e_mag,omega,theta] = cartesian2OrbitElements(r, v)

% r,v are 3x1 vectors
 r = reshape(r, [3,1]);
 v = reshape(v, [3,1]);

 r_mag = norm(r);
 v_mag = norm(v);

 v_r = v' * r / r_mag;

 if (v_r > 0)
     fprintf('v_r = %f > 0: satellite is flying away from perigee \n', v_r);
 end

 % step 4
 h = cross(r,v);

 %step 5
 h_mag = norm(h);

 %step 6
 i = acos( h(3)/h_mag );
 if(i > pi/2)
 fprintf("i = %f deg > 90 deg: retrograde orbit\n", rad2deg(i));
 else
     fprintf("Not in retrograde orbit\n");
 end

 K = [0 0 1]';
 N = cross(K,h);
 N_mag = norm(N);

 % 9 RAAN
 OMEGA = acos( N(1) / N_mag );

 if(N(2) < 0 )
     OMEGA = 2*pi - OMEGA;
 end
```

```
 % 10 Eccentricity vector
 mu = 398600;
 e = 1/mu * [ (v_mag^2 - mu / r_mag) * r - r_mag * v_r * v ];
 e_mag = norm(e);

 % 11
 omega = acos( N' * e / ( norm(N)*norm(e) ) );

 if (e(3) < 0)
     omega = 2*pi - omega;
 end

 % 13 True anomaly
 theta = acos( e' * r / ( e_mag * r_mag ) );

 if (v_r < 0)
     theta = 2*pi - theta;
 end
end
```

```
function [r_X,v_X] = orbit2GeocentricCartesian(h, e, i, OMEGA, omega, theta)

% state vectors r and v in geocentric equatorial frame: Page 175

    mu = 398600;

    p = h^2 / mu; % semi-latus rectum

    r_x = p / (1 + e * cos(theta) ) * [cos(theta) sin(theta) 0]';

    v_x = mu / h * [-sin(theta) e+cos(theta) 0]' ;

    QXx = C3(omega) * C1(i) * C3(OMEGA);
    QxX = QXx';

    r_X = QxX * r_x;
    v_X = QxX * v_x;

end
function R = C3(angle)

R = [ cos(angle) sin(angle) 0;
      -sin(angle) cos(angle) 0;
    0 0 1  ] ;
end
function R = C1(angle)
    R = [1 0 0;
        0 cos(angle) sin(angle);
        0 -sin(angle) cos(angle)];
end
```