SPACE FLIGHT MECHANICS

Project - 2

MAYANK AHUJA

AIM:

This report aims to Analyse the orbital life times for circular orbits of different sizes and also analyse the life times for different ballistic coefficients. This report also aims to Design Hohmann and Bielliptic transfers to circular orbits and uses comparison charts and tables for different intermediate orbit radii for bi elliptic transfers.

METHODOLOGY:

Simple lists were used from Python in the first question and for different values of ballistic coefficients respective orbital life times were computed. The CIRA Model density data was used for heights varying from 120 Km to 900 Km. In the second question the principles for Hohmann and bi-elliptic transfers were implemented. In both the questions numpy library was used predominantly.

QUESTION 1: Analyse the orbital life times for circular orbits of sizes varying from 200 km to 1000 km. Also analyse the life times for different ballistic coefficients varying from 50 to 300 kg/m 2 . You can use US standard/CIRA /exponential models for atmosphere. The CIRA model density data is attached for some

altitudes upto 900 km. You interpolate for other altitudes. You can use if a better model is available.

For, different values of h and ballistic coefficients orbital life times (in years) I got are shown in the table:

In the following table, Orbital lifetime (in years) is shown w.r.t the altitude and Ballistic coefficients.

h / B	50	100	150	200	250	300
200	0.0021	0.00421	0.0063	0.0084	0.0105	0.0126
260	0.01296	0.0259	0.0388	0.0518	0.0648	0.0777
320	0.0699	0.1398	0.2097	0.2797	0.3496	0.4195
380	0.2155	0.4310	0.6465	0.8620	1.0776	1.2931
440	0.7503	1.500	2.2509	3.0013	3.7516	4.5019
500	1.878	3.7562	5.6343	7.5124	9.3905	11.2686
560	5.478	10.9571	16.4357	21.9142	27.3928	32.8714
620	12.320	24.6400	36.9600	49.2800	61.6001	73.9201
680	32.039	64.0781	96.1172	128.1563	160.1954	192.2345
740	65.639	131.2787	196.9181	262.5575	328.1969	393.8363
800	150.0951	300.190	450.2853	600.3804	750.4756	900.5707
860	175.250	350.5014	525.7521	701.0028	876.2536	1051.5043
880	262.360	524.7218	787.0828	1049.4437	1311.8046	1574.1656

Question 2. Assume that a spacecraft is in 1000 km circular orbit around Earth. Design Hohmann and Bielliptic transfers to circular orbits of sizes 125000 km, 200000km and 300000 km. Make comparison charts and tables for different intermediate orbit radii for bi elliptic transfers (you can vary between 150000 km and 500000 km).

Hohmann transfer for circular orbits of 125000 km, 200000km and 300000 km is implemented starting from 1000 km circular orbit around Earth.

The final orbital radius for all three cases comes out as : [131378, 206378, 306378]

The different impulses given and the corresponding velocities at such points before and after impulses are given as:

Velocity at A in initial orbit comes out as : `7.350202797216279 Km/s`

Velocity at A in First transfer orbit for all 3 final radii: [10.11462432232843 10.213788541205572 10.27181261588848]
Velocity impulses(delta V(A)) at A in First transfer orbit for all 3 final radii: [2.7644215251121507, 2.863585743989293, 2.9216098186722013]
Velocity at B in second transfer orbit for all 3 final radii: [0.5680227911076374, 0.36514227222385415, 0.24735925386295732]

Velocity at B in final orbit for all 3 final radii: [1.741836248014132, 1.3897508472590643, 1.1406170490041625]
Velocity impulses(delta V(A)) at B in First transfer orbit for all 3 final radii: [1.1738134569064946, 1.02460857503521, 0.8932577951412052]
Total velocity impulse (delta_Vnet): [3.9382349820186455, 3.888194319024503, 3.8148676138134063]

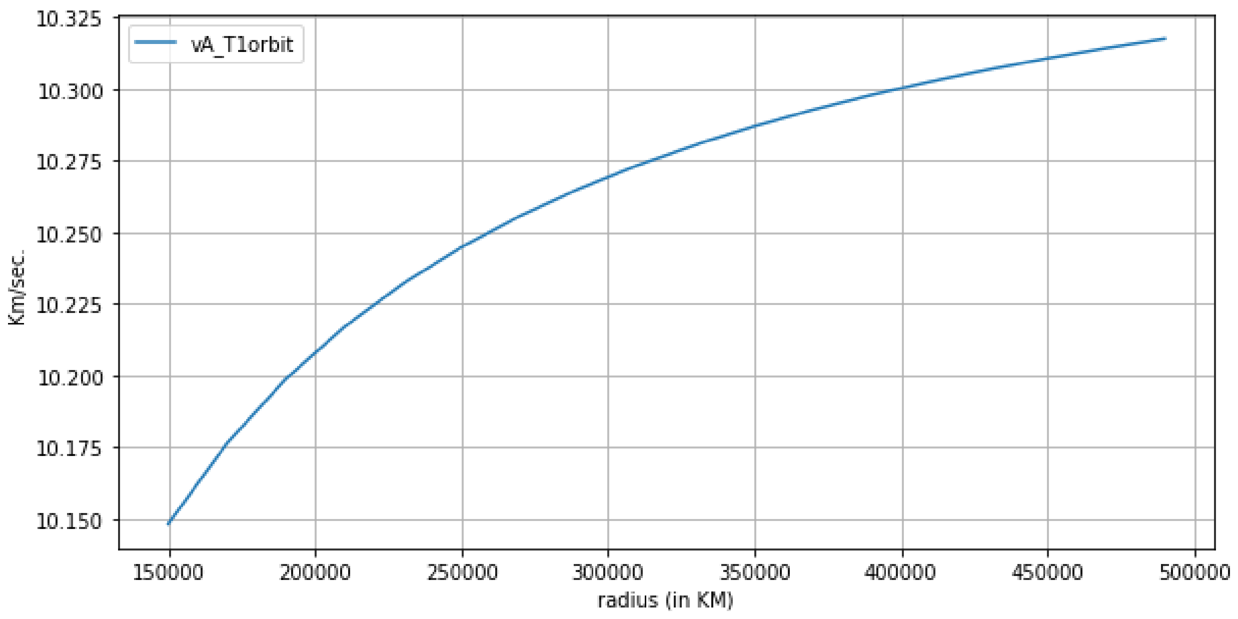
Now, coming to the Bielliptic transfers. The radii for final orbits come out same as: [131378, 206378, 306378]

The values of different radii between 150000 Km and 500000 Km are taken as: [150000, 170000, 190000, 210000, 230000, 250000, 270000, 290000, 310000, 333000, 350000, 370000, 390000, 410000, 430000, 450000, 470000, 490000]

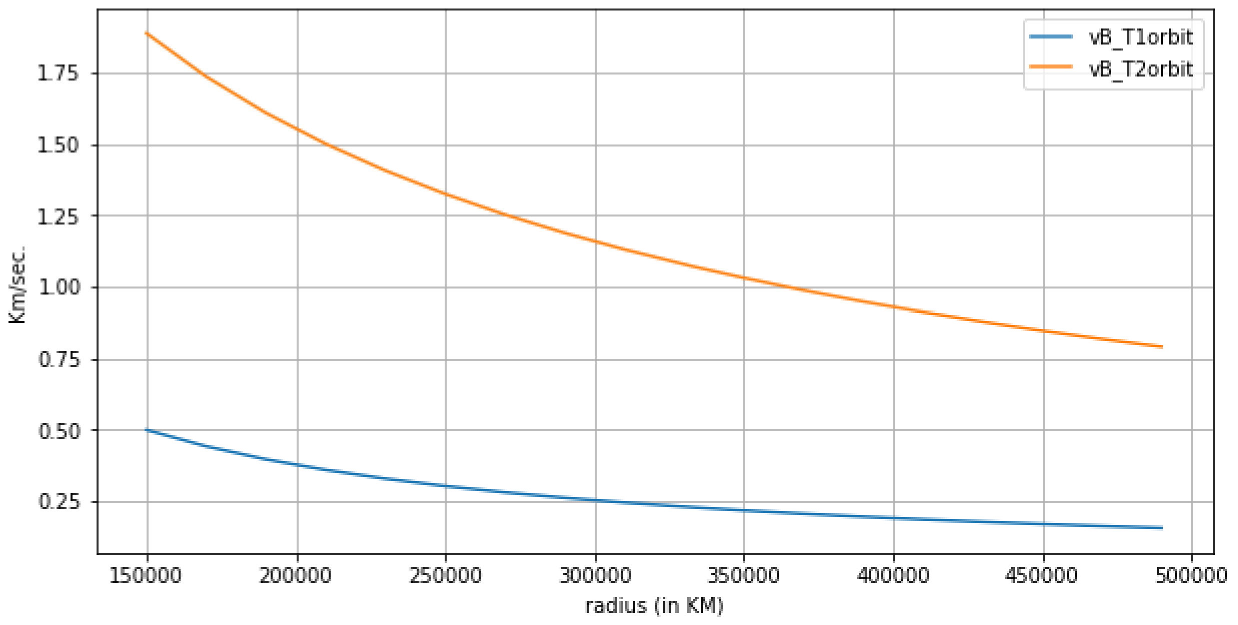
For final radii of 131378 Km, I got the following plots:

Velocity at A in initial orbit comes out as : `7.350202797216279 Km/s`

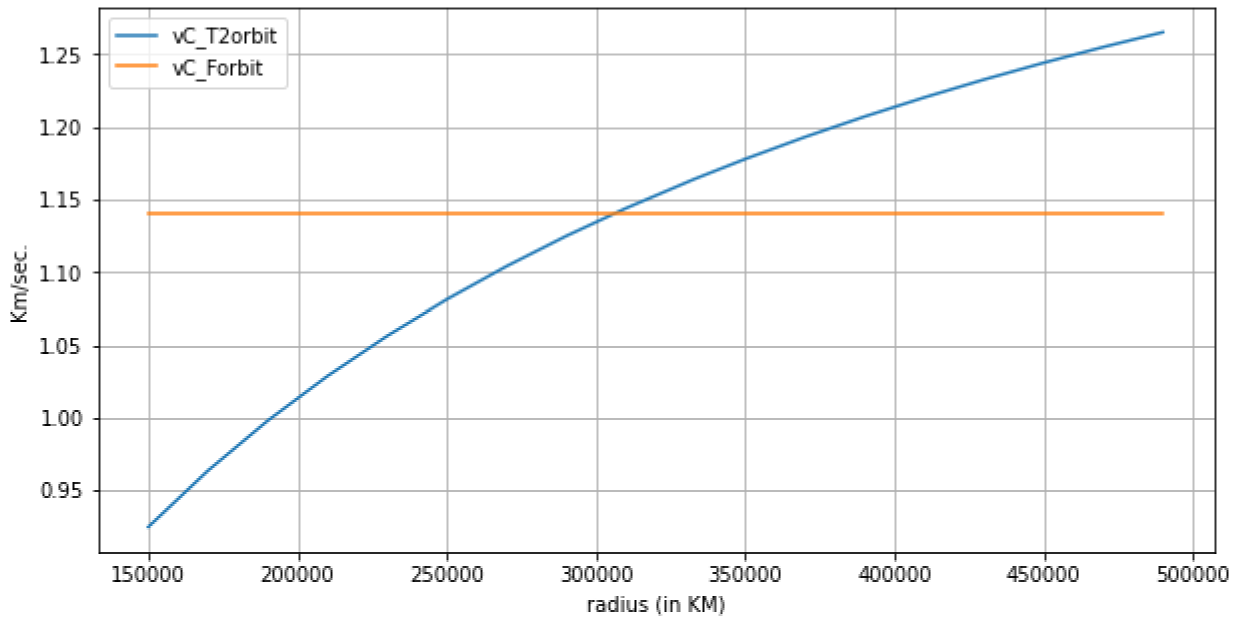
Plot of velocity at A in T1(First transfer orbit) comes out as:



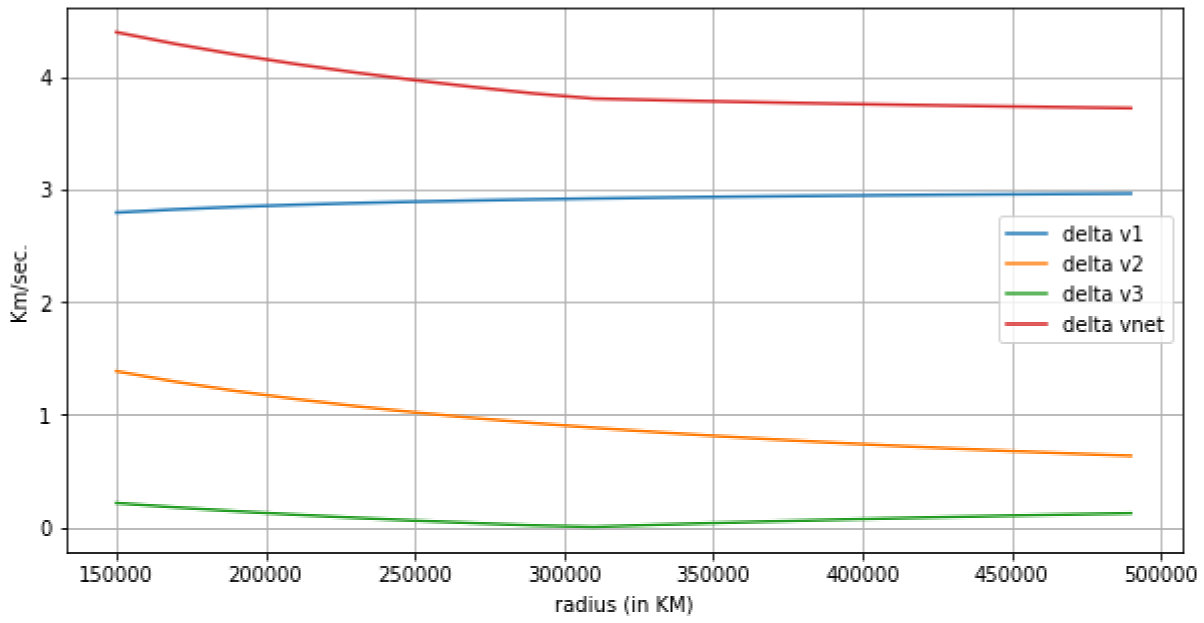
Plot of velocity at B in T1(First transfer orbit) and T2(Second transfer orbit) comes out as:



Plot of velocity at C in T2(second transfer orbit) and F(Final orbit) comes out as:



Finally, plot of delta v1, delta v2, delta v3 and delta vnet:



Similar plots would come for the rest of the other two final radii as well.

From the next page is shown the code for both the questions:

```
import numpy as np
import math
import matplotlib.pyplot as plt

mu=398600
Re=6378
B=300    #this can be varied
h_0=250  #this can be varied

f=[0]*39
h=[120,140,160,180,200,220,240,260,280,300,320,340,360,380,400,420,440,460
,480,500,520,540,560,580,600,620,640,660,680
   ,700,720,740,760,780,800,820,840,860,880]
density=[2.03E-08,3.44E-09,1.20E-09,5.46E-10,2.84E-10,1.61E-10,9.60E-11,5.
97E-11,3.83E-11,2.52E-11,1.69E-11,1.16E-11,

7.99E-12,5.60E-12,3.96E-12,2.83E-12,2.03E-12,1.47E-12,1.07E-12,7.85E-13,5.
78E-13,4.29E-13,3.19E-13,2.39E-13,

1.80E-13,1.36E-13,1.04E-13,7.98E-14,6.16E-14,4.80E-14,3.76E-14,2.98E-14,2.
38E-14,1.92E-14,1.57E-14,1.29E-14,
    1.07E-14,9.03E-15,7.67E-15]

for i in range(38):         #this can be varied for different cases of h_0
f[i]=1/(density[i]*np.sqrt(mu*(h[i]+Re)))

x=0
y=0

for i in range(17):
 x=x+f[2*(i+1)]
 y=y+f[1+2*i]

I=((20/3)*(f[0]+f[38]))+((40/3)*(x)+((80/3)*(y+f[37])))
print(I)
T=B*(10**(-3))*I
print(T)
Timein_H=T/3600
Timein_D=Timein_H/24
Timein_Y=Timein_D/365.25
```

```python
print('orbital lifetime in days is ', Timein_D)
print('orbital lifetime in years is, ', Timein_Y)
```

For hohmann transfer:

```python
import numpy as np
import matplotlib.pyplot as plt
mu=398600
a0=7378
a=[131378, 206378, 306378]

vA_Initial=np.sqrt(mu/a0)
print( a)
print(vA_Initial)
vA_T1=[0]*3
delta_v1=[0]*3
a_1=[0]*3
for i in range(3):
a_1[i]=(a[i]+a0)/2
vA_T1[i]=np.sqrt(mu*(2/a0-1/a_1[i]))
delta_v1[i] = np.abs(vA_T1[i]-vA_Initial)
print(vA_Torbit)
print(delta_v1)

vB_T1=[0]*3
vB_F=[0]*3
delta_v2=[0]*3
delta_Vnet=[0]*3

for i in range(3):
   vB_T1[i]=np.sqrt(mu*(2/a[i]-1/a_1[i]))
   vB_F[i]=np.sqrt(mu/a[i])
   delta_v2[i]=np.abs(vB_F[i]-vB_T1[i])
   delta_Vnet[i]=delta_v2[i]+delta_v1[i]
print(vB_Torbit)
print(vB_Forbit)
print(delta_v2)
print(delta_Vnet)
```

For bi-elliptic transfer:

```
import numpy as np
import matplotlib.pyplot as plt

mu=398600
r=[150000, 170000, 190000, 210000, 230000, 250000, 270000, 290000, 310000,
333000, 350000, 370000, 390000, 410000, 430000, 450000, 470000, 490000]
a0=7378
vA_Iorbit=np.sqrt(mu/a0)
print(a)
print(vA_Iorbit)

vA_T1orbit=[[0]*18]*3
delta_v1=[[0]*18]*3
a_1=[131378, 206378, 306378]
b1=[[0]*18]*3
b2=[[0]*18]*3


for i in range(3):
   for j in range(18):
    b1[i][j]=(r[j]+a0)/2
    vA_T1orbit[i][j]=np.sqrt(mu*(2/a0-1/b1[i][j]))
    delta_v1[i][j]=np.abs(vA_T1orbit[i][j]-vA_Iorbit)


print(vA_T1orbit)
print(delta_v1)

vB_T1orbit=[[0]*18]*3
vB_T2orbit=[[0]*18]*3
delta_v2=[[0]*18]*3


for i in range(3):
   for j in range(18):
       b2[i][j]=(r_c[j]+a_1[i])/2
       vB_T1orbit[i][j]=np.sqrt(mu*(2/r_c[j]-1/b1[i][j]))
       vB_T2orbit[i][j]=np.sqrt(mu*(2/r_c[j]-1/b2[i][j]))
       delta_v2[i][j]=np.abs(vB_T1orbit[i][j]-vB_T2orbit[i][j])

print(vB_T1orbit)
print(vB_T2orbit)
print(delta_v2)

vC_T2orbit=[[0]*18]*3
vC_Forbit=[[0]*18]*3
delta_v3=[[0]*18]*3
delta_Vnet=[[0]*18]*3
for i in range(3):
   for j in range(18):

       vC_T2orbit[i][j]=np.sqrt(mu*(2/a_1[i]-1/b2[i][j]))
       vC_Forbit[i][j]=np.sqrt(mu/a_1[i])
       delta_v3[i][j]=np.abs(vC_T2orbit[i][j]-vC_Forbit[i][j])
       delta_Vnet[i][j]=delta_v1[i][j]+delta_v2[i][j]+delta_v3[i][j]

print(vC_T2orbit)
print(vC_Forbit)
print(delta_v3)
print(delta_Vnet)

plt.figure(figsize=(10,5))
plt.grid()
plt.plot(r_c,vA_T1orbit[1],label='vA_T1orbit')

plt.xlabel('radius (in KM)')
plt.ylabel('Km/sec.')
plt.legend(loc='best')
plt.show()

plt.figure(figsize=(10,5))
plt.grid()
plt.plot(r_c,vB_T1orbit[1],label='vB_T1orbit')
plt.plot(r_c,vB_T2orbit[1],label='vB_T2orbit')
plt.xlabel('radius (in KM)')
plt.ylabel('Km/sec.')
plt.legend(loc='best')
plt.show()

plt.figure(figsize=(10,5))
plt.grid()
plt.plot(r_c,delta_v1[1],label='delta v1')
plt.plot(r_c,delta_v2[1],label= 'delta v2')
plt.plot(r_c,delta_v3[1],label= 'delta v3')
plt.plot(r_c,delta_Vnet[1],label= 'delta vnet')
plt.legend(loc='best')
plt.xlabel('radius (in KM)')
plt.ylabel('Km/sec.')
plt.legend(loc='best')
plt.show()

plt.figure(figsize=(10,5))
plt.grid()
plt.plot(r_c,vC_T2orbit[1],label='vC_T2orbit')
plt.plot(r_c,vC_Forbit[1],label= 'vC_Forbit')
plt.xlabel('radius (in KM)')
plt.ylabel('Km/sec.')
plt.legend(loc='best')
plt.show()
```