Function:

function [dv1,dv2,dv3,posT,time] = biEllipticWithPlaneChange(oei,oef,S,f,mu)

%This function calculates the 3 impulses needed to complete a bi-elliptic orbit transfer

%Input: oei, vector containing the orbital elements of the initial orbit

%Input: oef, vector containing the orbital elements of the final orbit

%Input: S, the ratio ri/rf. The ratio of apoapsis of transfer orbit to radius of final orbit

%Input: f, fraction of orbit crank performed at first impulse

%Input: mu, gravitaional constant

%Output: dv1, magnitude of impulse one

%Output: dv2, magnitude of impulse two

%Output: dv3, magnitude of impulse three

%Output: posT, matrix containing position of transfer orbit

%Output: time, vector containing times associated with postitions in posT

[ri,vi] = oe2rv\_Hackbardt\_Chris(oei,mu);

[rf,vf] = oe2rv\_Hackbardt\_Chris(oef,mu);

hvec1=cross(ri,vi);

hvec2=cross(rf,vf);

lvec=cross(hvec1,hvec2);

lvec=lvec/norm(lvec);

u1vec=cross(hvec1,lvec)/norm(hvec1);

u2vec=cross(hvec2,lvec)/norm(hvec2);

r0=oei(1);

rf=oef(1);

ra=S\*rf;

v0=sqrt(mu/r0);

vf=sqrt(mu/rf);

posdV1=r0\*lvec;

veldV1b4=v0\*u1vec;

veldV3after=vf\*u2vec;

%First Transfer Orbit

%Uses Rodrigues' rotation formula to find vel vector at periapsis of first transfer orbit

at1=(r0+ra)/2;

vpT1=sqrt(mu)\*sqrt((2/r0)-(1/at1));

dv1afterDirection=angleBetweenVectors(hvec1,hvec2)\*f;

I=eye(3);

K=[0,-lvec(3),lvec(2);lvec(3),0,-lvec(1);-lvec(2),lvec(1),0];

R=I+(sin(dv1afterDirection).\*K)+((1-cos(dv1afterDirection)).\*K^2);

vel1T1=R\*veldV1b4;

vel1T1=(vel1T1/norm(vel1T1))\*vpT1;

oet1=rv2oe\_Hackbardt\_Chris(posdV1,vel1T1,mu);

tauT1=2\*pi\*sqrt(oet1(1)^3/mu);

timeT1=0:60:tauT1/2;

for i=1:length(timeT1)

[r,v] = propagateKepler\_Hackbardt\_Chris(posdV1,vel1T1,0,timeT1(i),mu);

posT1(i,:)=r';

velT1(i,:)=v';

end

%Second Transfer Orbit

%Uses Rodrigues' rotation formula to find vel vector at apoapsis transfer orbit

at2=(rf+ra)/2;

vAT2=sqrt(mu)\*sqrt((2/ra)-(1/at2));

hvecT1=cross(posdV1,vel1T1);

dv2afterDirection=-angleBetweenVectors(hvecT1,hvec2);

posdV2=posT1(end,:);

posDV2dir=posdV2/norm(posdV2);

K=[0,-posDV2dir(3),posDV2dir(2);posDV2dir(3),0,-posDV2dir(1);-posDV2dir(2),posDV2dir(1),0];

R=I+(sin(dv2afterDirection).\*K)+((1-cos(dv2afterDirection)).\*K^2);

vellastT1=velT1(end,:)';

vel1T2dir=R\*(vellastT1/norm(vellastT1));

vel1T2=vel1T2dir\*vAT2;

oet2=rv2oe\_Hackbardt\_Chris(posdV2,vel1T2,mu);

tauT2=2\*pi\*sqrt(oet2(1)^3/mu);

timeT2=0:60:tauT2/2;

for i=1:length(timeT2)

[r,v] = propagateKepler\_Hackbardt\_Chris(posdV2,vel1T2,0,timeT2(i),mu);

posT2(i,:)=r';

velT2(i,:)=v';

end

posT=[posT1;posT2];

timeT2=timeT2+timeT1(end);

vellastT2=velT2(end,:);

%Finds the delta vs

dv1=norm(vel1T1-veldV1b4);

dv2=norm(vel1T2-vellastT1);

dv3=norm(veldV3after-vellastT2');

time=[timeT1';timeT2'];

end

Main:

clc;clear;close all;

mu=398600;

S=[2,5,10,20];

f=[0,0.5,1];

r1=350+6378.145;

i1=deg2rad(28);

Omega1=deg2rad(60);

oei=[r1,0,Omega1,i1,0,0];

r2=20200+6378.145;

i2=deg2rad(57);

Omega2=deg2rad(120);

oef=[r2,0,Omega2,i2,0,0];

[pos1,v1] = oe2rv\_Hackbardt\_Chris(oei,mu);

tau1=2\*pi\*sqrt(r1^3/mu);

[times1, pos1, vel1] = propagateOnCircle(pos1,v1,0,tau1,mu,500);

[pos2,v2] = oe2rv\_Hackbardt\_Chris(oef,mu);

tau2=2\*pi\*sqrt(r2^3/mu);

[times2, pos2, vel2] = propagateOnCircle(pos2,v2,0,tau2,mu,500);

R=r2/r1;

dVBP=norm(v1)\*((sqrt(2)-1)+sqrt(1/R)\*(sqrt(2)-1));

dvp=zeros(1,length(f));

dvp(:)=dVBP;

totdV=zeros(length(S),length(f));

for i=1:length(f)

figure

hold on

for j=1:length(S)

[dv1,dv2,dv3,posT,time]=biEllipticWithPlaneChange(oei,oef,S(j),f(i),mu);

plot3(posT(:,1),posT(:,2),posT(:,3))

totdV(j,i)=dv1+dv2+dv3;

end

plot3(pos1(:,1),pos1(:,2),pos1(:,3))

plot3(pos2(:,1),pos2(:,2),pos2(:,3))

view(-160,55)

end

figure

hold on

plot(f,totdV(1,:),f,totdV(2,:),f,totdV(3,:),f,totdV(4,:),f,dvp)

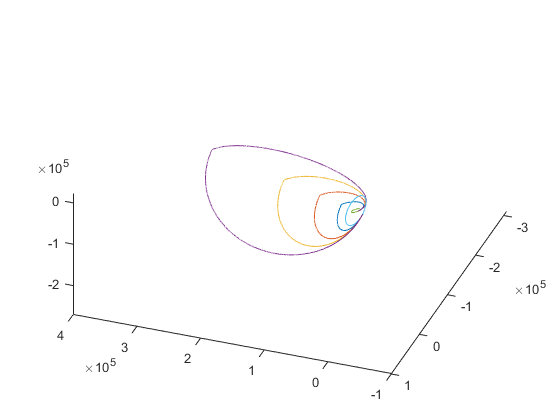
xlabel('f')

ylabel('dV')

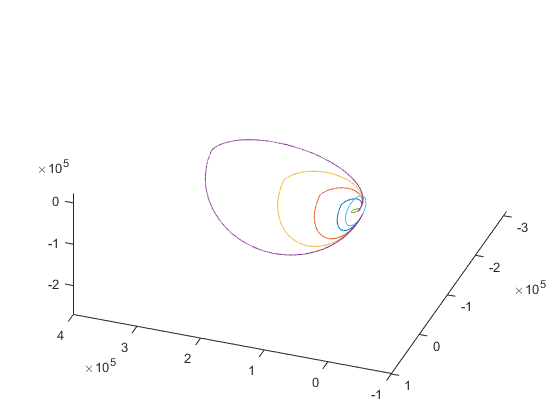
legend('S = 2','S = 5','S = 10','S = 20','Bi-Parabolic','Location','northwest')

3D plots:

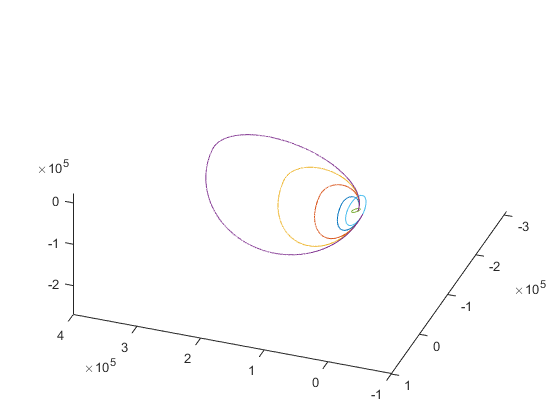
f = 0:



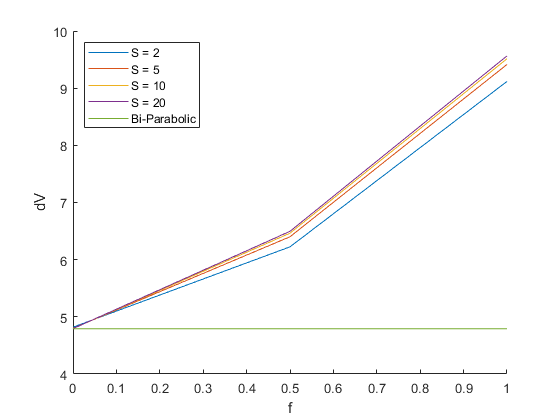
f = 0.5:



f = 1:



f vs ΔV:



The minimum ΔV is used for every value of S when f = 0. This is because it is more efficient to change the orbit plane at apoapsis of the transfer orbit instead of periapsis.