Problem 2

```
interr = 'latex';
% interr = 'none';
set(groot,'defaulttextinterpreter',interr);
set(groot, 'defaultAxesTickLabelInterpreter',interr);
set(groot, 'defaultLegendInterpreter',interr);
```

Preliminary Calculations and Constants:

```
Rm = 1738.2; % Lunar radius, km
mu = 4902.8005821478; % Lunar mu, km^3/s^2
```

Simple Plane Change:

Initial Orbit Characteristics:

```
om0 = 0; inc0 = 0; th0 = 0; r0 = Rm + 100, v0mag = sqrt(mu/r0)
```

```
r0 = 1.8382e+03
v0mag = 1.6331
```

```
iCr0 = findDCM(om0,inc0,th0)
```

```
iCr0 = 3×3
     1     0     0
     0     1     0
     0     0     1
```

```
v0_R = [0 v0mag 0]';
v0_I = iCr0*v0_R
```

```
v0_I = 3×1
         0
    1.6331
         0
```

Final Orbit Characteristics:

```
omN = 0; incN = deg2rad(90); thN = 0; rN = r0, vNmag = v0mag
```

```
rN = 1.8382e+03
vNmag = 1.6331
```

```
iCrN = findDCM(omN,incN,thN)
```

```
iCrN = 3×3
    1.0000         0         0
         0    0.0000   -1.0000
         0    1.0000    0.0000
```

```
vN_R = [0 vNmag 0]';
vN_I = iCrN*vN_R
```

```
vN_I = 3×1
         0
    0.0000
```

```
    1.6331
```

```
dv = vN_I - v0_I
```

```
dv = 3×1
         0
   -1.6331
    1.6331
```

```
dvmag = norm(dv)
```

```
dvmag = 2.3096
```

Bi-Elliptic Plane Change:

Initial Oribit @ point 1:

```
om1_0 = 0; inc1_0 = 0; th1_0 = 0; iCr1_0 =  findDCM(om1_0,inc1_0,th1_0)
```

```
iCr1_0 = 3×3
     1     0     0
     0     1     0
     0     0     1
```

```
r1mag_0 = 100+Rm;
v1mag_0 = sqrt(mu/r1mag_0), v1_0_R= [0 v1mag_0 0]'
```

```
v1mag_0 = 1.6331
v1_0_R = 3×1
         0
    1.6331
         0
```

```
v1_0 = iCr1_0 * v1_0_R
```

```
v1_0 = 3×1
         0
    1.6331
         0
```

1st Trasnfer Orbit @ point 1:

```
om1_n = 0; inc1_n = 0; th1_n = 0; iCr1_n =  findDCM(om1_n,inc1_n,th1_n)
```

```
iCr1_n = 3×3
     1     0     0
     0     1     0
     0     0     1
```

```
r1mag_n = r1mag_0; rp = r1mag_n; ra = 17500;
aT1 = (rp + ra)/2
```

```
aT1 = 9.6691e+03
```

```
v1mag_n = sqrt(mu*(2/r1mag_n - 1/aT1)), v1_n_R = [0 v1mag_n 0]'
```

```
v1mag_n = 2.1971
v1_n_R = 3×1
```

```
         0
    2.1971
         0
```

```
v1_n = iCr1_n * v1_n_R
```

```
v1_n = 3×1
         0
    2.1971
         0
```

```
dv1 = v1_n - v1_0, dv1mag = norm(dv1)
```

```
dv1 = 3×1
         0
    0.5640
         0
dv1mag = 0.5640
```

1st Transfer Orbit @ point 2:

```
om2_0 = 0; inc2_0 = 0; th2_0 = pi; iCr2_0 = findDCM(om2_0,inc2_0,th2_0)
```

```
iCr2_0 = 3×3
   -1.0000   -0.0000         0
    0.0000   -1.0000         0
         0         0    1.0000
```

```
r2mag_0 = ra;
v2mag_0 = sqrt(mu*(2/r2mag_0 - 1/aT1)), v2_0_R = [0 v2mag_0 0]'
```

```
v2mag_0 = 0.2308
v2_0_R = 3×1
         0
    0.2308
         0
```

```
v2_0 = iCr2_0 * v2_0_R
```

```
v2_0 = 3×1
   -0.0000
   -0.2308
         0
```

2nd Transfer Orbit @ point 2:

```
om2_n = 0; inc2_n = -pi/2; th2_n = pi; iCr2_n = findDCM(om2_n,inc2_n,th2_n)
```

```
iCr2_n = 3×3
   -1.0000   -0.0000         0
    0.0000   -0.0000    1.0000
   -0.0000    1.0000    0.0000
```

```
r2mag_n = ra;
aT2 = aT1;
v2mag_n = sqrt(mu*(2/r2mag_n - 1/aT2)), v2_n_R = [0 v2mag_n 0]'
```

```
v2mag_n = 0.2308
v2_n_R = 3×1
```

```
         0
    0.2308
         0
```

```
v2_n = iCr2_n * v2_n_R
```

```
v2_n = 3×1
   -0.0000
   -0.0000
    0.2308
```

```
dv2 = v2_n - v2_0, dv2mag = norm(dv2)
```

```
dv2 = 3×1
         0
    0.2308
    0.2308
dv2mag = 0.3264
```

2nd Transfer Orbit @ point 3:

```
om3_0 = 0; inc3_0 = -pi/2; th3_0 = 0; iCr3_0 = findDCM(om3_0,inc3_0,th3_0)
```

```
iCr3_0 = 3×3
    1.0000         0         0
         0    0.0000    1.0000
         0   -1.0000    0.0000
```

```
r3mag_0 = rp;
v3mag_0 = sqrt(mu*(2/r3mag_0 - 1/aT2)), v3_0_R = [0 v3mag_0 0]'
```

```
v3mag_0 = 2.1971
v3_0_R = 3×1
         0
    2.1971
         0
```

```
v3_0 = iCr3_0 * v3_0_R
```

```
v3_0 = 3×1
         0
    0.0000
   -2.1971
```

Final Orbit @ point 3:

```
om3_n = 0; inc3_n = -pi/2; th3_n = 0; iCr3_n = findDCM(om3_n,inc3_n,th3_n)
```

```
iCr3_n = 3×3
    1.0000         0         0
         0    0.0000    1.0000
         0   -1.0000    0.0000
```

```
r3mag_n = rp;
v3mag_n = sqrt(mu/r3mag_n), v3_n_R = [0 v3mag_n 0]'
```

```
v3mag_n = 1.6331
v3_n_R = 3×1
         0
```

```
    1.6331
         0
```

```
v3_n = iCr3_n * v3_n_R
```

```
v3_n = 3×1
         0
    0.0000
   -1.6331
```

```
dv3 = v3_n - v3_0, dv3mag = norm(dv3)
```

```
dv3 = 3×1
         0
   -0.0000
    0.5640
dv3mag = 0.5640
```

```
dvtotal = dv1mag + dv2mag + dv3mag
```

```
dvtotal = 1.4543
```

```
TOF_bielliptic = 2*pi*sqrt(aT1^3/mu), TOF_hr = TOF_bielliptic/3600
```

```
TOF_bielliptic = 8.5317e+04
TOF_hr = 23.6992
```

```
1-dvtotal/dvmag*100
```

```
ans = -61.9671
```

Function 3: Find Direction Cosine Matrix

```
function iCr = findDCM(om,inc,th)
    col1 = [cos(om)*cos(th) - sin(om)*cos(inc)*sin(th);
        sin(om)*cos(th)+cos(om)*cos(inc)*sin(th);
        sin(inc)*sin(th)];
%     col2 = [-cos(om)*sin(th)-sin(om)*cos(inc)*cos(th);
%         -sin(om)*sin(th)+cos(om)*cos(inc)*cos(th);
%         sin(inc)*cos(th)];
    col3 = [sin(om)*sin(inc);
        -cos(om)*sin(inc);
        cos(inc)];

    col2 = cross(col3,col1);
    iCr = [col1 col2 col3];
end
```