```
 clear; close all; clc;

mue  = 398600 ;
muven = 324900 ;
re = 6378 ;
rven = 6052 ;
mus = 1.32712e11 ;
opts = odeset( 'AbsTol' , 1e-8 , 'RelTol' , 1e-8 ) ;
d2s = 24*60*60 ;
rsun = 696000 ;
disp( ' ' )
        disp( '2' )
        k = 4 ;
for jj = 1:24*k
        % Planetary states
            dateDeparture = [ 1 , 12 , 2018 ] ;
            timeDeparture(jj,:) = [ jj/k , 0 , 0 ] ;
            [ ~ , ~ , ~ , JdDeparture(jj) ] =
 Julian( timeDeparture(jj,:) , dateDeparture ) ;
            tDeparture(jj) = JdDeparture(jj)/365.25 ;
            coesEarth = planetary_elements( 3 , tDeparture(jj) ) ;
            [ rE(:,jj) , vE(:,jj) ] = Pcoes2state( coesEarth , mus ) ;

            dateArr1 = [ 1 , 3 , 2019 ] ;
            timeArr1(jj,:) = [ jj/k , 0 , 0 ] ;
            [ ~ , ~ , ~ , JdArr1(jj) ] = Julian( timeArr1(jj,:) ,
 dateArr1 ) ;
            tArr1(jj) = JdArr1(jj)/365.25 ;
            coesVenus1 = planetary_elements( 2 , tArr1(jj) ) ;
            [ rV1(:,jj) , vV1(:,jj) ] = Pcoes2state( coesVenus1 ,
 mus ) ;

            dateArr2 = [ 1 , 4 , 2019 ] ;
            timeArr2(jj,:) = [ jj/k , 0 , 0 ] ;
            [ ~ , ~ , ~ , JdArr2(jj) ] = Julian( timeArr2(jj,:) ,
 dateArr2 ) ;
            tArr2(jj) = JdArr2(jj)/365.25 ;
            coesVenus2 = planetary_elements( 2 , tArr2(jj) ) ;
            [ rV2(:,jj) , vV2(:,jj) ] = Pcoes2state( coesVenus2 ,
 mus ) ;

            dateArr3 = [ 1 , 5 , 2019 ] ;
            timeArr3 = [ jj/k , 0 , 0 ] ;
            [ ~ , ~ , ~ , JdArr3(jj) ] = Julian( timeArr3 ,
 dateArr3 ) ;
            tArr3(jj) = JdArr3(jj)/365.25 ;
            coesVenus3 = planetary_elements( 2 , tArr3(jj) ) ;
            [ rV3(:,jj) , vV3(:,jj) ] = Pcoes2state( coesVenus3 ,
 mus ) ;
end
 kk = 0 ;
 for ii = 1:24*k
```

```
    for jj = 1:24*k
        % Interplanetary
        kk = kk + 1 ;
        progress = 100*kk/(24*k)^2 ;
            dt1(kk) = ( -JdDeparture(ii) + JdArr1(jj) )*d2s;

            [ vDep1(:,kk) , vArr1(:,kk) ] = Lamberts2( rE(:,ii) ,
 rV1(:,jj) , dt1(kk) , mus , 1e-5 , 1 ) ;
            [ vDep1r(:,kk) , vArr1r(:,kk) ] = Lamberts2( rE(:,ii) ,
 rV1(:,jj) , dt1(kk) , mus , 1e-5 , 0 ) ;

    end
 end
kk = 0 ;
for ii = 1:24*k
    for jj = 1:24*k
        % Interplanetary
        kk = kk + 1 ;
        progress2 = 100*kk/(24*k)^2 ;
            dt2(kk) = ( -JdDeparture(ii) + JdArr2(jj) )*d2s;

            [ vDep2(:,kk) , vArr2(:,kk) ] = Lamberts2( rE(:,ii) ,
 rV2(:,jj) , dt2(kk) , mus , 1e-5 , 1 ) ;
            [ vDep2r(:,kk) , vArr2r(:,kk) ] = Lamberts2( rE(:,ii) ,
 rV2(:,jj) , dt2(kk) , mus , 1e-5 , 0 ) ;

    end
end
kk = 0 ;
for ii = 1:24*k
    for jj = 1:24*k
        % Interplanetary
        kk = kk + 1 ;
        progress3 = 100*kk/(24*k)^2 ;
            dt3(kk) = ( -JdDeparture(ii) + JdArr3(jj) )*d2s;

            [ vDep3(:,kk) , vArr3(:,kk) ] = Lamberts2( rE(:,ii) ,
 rV3(:,jj) , dt3(kk) , mus , 1e-5 , 1 ) ;
            [ vDep3r(:,kk) , vArr3r(:,kk) ] = Lamberts2( rE(:,ii) ,
 rV3(:,jj) , dt3(kk) , mus , 1e-5 , 0 ) ;

    end
end


            % Departure
            zDPark = 180 ;
            rDPark = re + zDPark ; % radius of departure park
            vDPark = sqrt( mue / rDPark ) ; % velocity of parking
 orbit of departure
            RsoiE = 925000 ;
            for ii = 1:length( vDep1 )
                vinfD = norm( vDep1(:,ii) - vE(:,ceil(ii/(24*k))) ) ;
                aHD = (mue/2)*((vinfD^2/2)-(mue/RsoiE))^(-1) ;
                eccHD = rDPark/aHD + 1 ;
```

```
                hHD = sqrt( rDPark*mue*(1+eccHD) ) ;
                vpHD = (mue/hHD)*(1+eccHD) ;
                dvD(ii) = norm( vpHD - vDPark ) ;

                vinfDr = norm( vDep1r(:,ii) - vE(:,ceil(ii/
(24*k))) ) ;
                aHD = (mue/2)*((vinfDr^2/2)-(mue/RsoiE))^(-1) ;
                eccHD = rDPark/aHD + 1 ;
                hHD = sqrt( rDPark*mue*(1+eccHD) ) ;
                vpHDr = (mue/hHD)*(1+eccHD) ;
                dvDr(ii) = norm( vpHDr - vDPark ) ;
            end
            for ii = 1:length( vDep2 )
                vinfD = norm( vDep2(:,ii) - vE(:,ceil(ii/(24*k))) ) ;
                aHD = (mue/2)*((vinfD^2/2)-(mue/RsoiE))^(-1) ;
                eccHD = rDPark/aHD + 1 ;
                hHD = sqrt( rDPark*mue*(1+eccHD) ) ;
                vpHD = (mue/hHD)*(1+eccHD) ;
                dvD2(ii) = norm( vpHD - vDPark ) ;

                vinfDr = norm( vDep2r(:,ii) - vE(:,ceil(ii/
(24*k))) ) ;
                aHD = (mue/2)*((vinfDr^2/2)-(mue/RsoiE))^(-1) ;
                eccHD = rDPark/aHD + 1 ;
                hHD = sqrt( rDPark*mue*(1+eccHD) ) ;
                vpHDr = (mue/hHD)*(1+eccHD) ;
                dvDr2(ii) = norm( vpHDr - vDPark ) ;
             end
            for ii = 1:length( vDep3 )
                vinfD = norm( vDep3(:,ii) - vE(:,ceil(ii/(24*k))) ) ;
                aHD = (mue/2)*((vinfD^2/2)-(mue/RsoiE))^(-1) ;
                eccHD = rDPark/aHD + 1 ;
                hHD = sqrt( rDPark*mue*(1+eccHD) ) ;
                vpHD = (mue/hHD)*(1+eccHD) ;
                dvD3(ii) = norm( vpHD - vDPark ) ;

                vinfDr = norm( vDep3r(:,ii) - vE(:,ceil(ii/
(24*k))) ) ;
                aHD = (mue/2)*((vinfDr^2/2)-(mue/RsoiE))^(-1) ;
                eccHD = rDPark/aHD + 1 ;
                hHD = sqrt( rDPark*mue*(1+eccHD) ) ;
                vpHDr = (mue/hHD)*(1+eccHD) ;
                dvDr3(ii) = norm( vpHDr - vDPark ) ;
             end

            % Arrival
            zAa = 10000 ;
            rAa = zAa + rven ;
            zAp = 200 ;
            rAp = zAp + rven ;
            eccA = ( rAa - rAp )/( rAp + rAp ) ;
            hA = sqrt( rAp*muven*(1+eccA) ) ;
            vpA = (muven/hA)*(1+eccA) ;
            RsoiV = 616000 ;
```

```
for ii = 1:length( vArr1 )
    vinfA = norm( vArr1(:,ii) - vV1(:,ceil(ii/(24*k))) ) ;
    aHA = (muven/2)*((vinfA^2/2)-(muven/RsoiV))^(-1) ;
    eccHA = rAp/aHA + 1 ;
    hHA = sqrt( rAp*muven*(1+eccHA) ) ;
    vpHA = (muven/hHA)*(1+eccHA) ;
    dvA(ii) = norm( vpHA - vpA ) ;

    vinfAr = norm( vArr1r(:,ii) - vV1(:,ceil(ii/
(24*k))) ) ;
    aHA = (muven/2)*((vinfA^2/2)-(muven/RsoiV))^(-1) ;
    eccHA = rAp/aHA + 1 ;
    hHA = sqrt( rAp*muven*(1+eccHA) ) ;
    vpHAr = (muven/hHA)*(1+eccHA) ;
    dvAr(ii) = norm( vpHAr - vpA ) ;
end
for ii = 1:length( vArr2 )
    vinfA = norm( vArr2(:,ii) - vV2(:,ceil(ii/(24*k))) ) ;
    aHA = (muven/2)*((vinfA^2/2)-(muven/RsoiV))^(-1) ;
    eccHA = rAp/aHA + 1 ;
    hHA = sqrt( rAp*muven*(1+eccHA) ) ;
    vpHA = (muven/hHA)*(1+eccHA) ;
    dvA2(ii) = norm( vpHA - vpA ) ;

    vinfAr = norm( vArr2(:,ii) - vV2(:,ceil(ii/
(24*k))) ) ;
    aHA = (muven/2)*((vinfA^2/2)-(muven/RsoiV))^(-1) ;
    eccHA = rAp/aHA + 1 ;
    hHA = sqrt( rAp*muven*(1+eccHA) ) ;
    vpHAr = (muven/hHA)*(1+eccHA) ;
    dvAr2(ii) = norm( vpHAr - vpA ) ;
end
for ii = 1:length( vArr3 )
    vinfA = norm( vArr3(:,ii) - vV3(:,ceil(ii/(24*k))) ) ;
    aHA = (muven/2)*((vinfA^2/2)-(muven/RsoiV))^(-1) ;
    eccHA = rAp/aHA + 1 ;
    hHA = sqrt( rAp*muven*(1+eccHA) ) ;
    vpHA = (muven/hHA)*(1+eccHA) ;
    dvA3(ii) = norm( vpHA - vpA ) ;

    vinfAr = norm( vArr3r(:,ii) - vV3(:,ceil(ii/
(24*k))) ) ;
    aHA = (muven/2)*((vinfA^2/2)-(muven/RsoiV))^(-1) ;
    eccHA = rAp/aHA + 1 ;
    hHA = sqrt( rAp*muven*(1+eccHA) ) ;
    vpHAr = (muven/hHA)*(1+eccHA) ;
    dvAr3(ii) = norm( vpHAr - vpA ) ;
end
dv = dvA + dvD ;
dv2 = dvA2 + dvD2 ;
dv3 = dvA3 + dvD3 ;
dvr = dvAr + dvDr ;
dvr2 = dvAr2 + dvDr2 ;
dvr3 = dvAr3 + dvDr3 ;
```

```
            [ dvr_best1 , r1 ] = min( dvr ) ;
            [ dvr_best2 , r2 ] = min( dvr2 ) ;
            [ dvr_best3 , r3 ] = min( dvr3 ) ;
            [ dv_best1 , p1 ] = min( dv ) ;
            [ dv_best2 , p2 ] = min( dv2 ) ;
            [ dv_best3 , p3 ] = min( dv3 ) ;
            planetpos1 = ceil(p1/(24*k)) ;
            planetpos2 = ceil(p2/(24*k)) ;
            planetpos3 = ceil(p3/(24*k)) ;
            planetpos1r = ceil(r1/(24*k)) ;
            planetpos2r = ceil(r2/(24*k)) ;
            planetpos3r = ceil(r3/(24*k)) ;
        [ t , stateT1 ] = ode45( @TwoBodyMotion , [ 0 dt1(p1) ] ,
 [ rE(:,planetpos1) ; vDep1(:,p1) ] , opts , mus ) ;
        [ t , stateT2 ] = ode45( @TwoBodyMotion , [ 0 dt2(p2) ] ,
 [ rE(:,planetpos2) ; vDep2(:,p2) ] , opts , mus ) ;
        [ t , stateT3 ] = ode45( @TwoBodyMotion , [ 0 dt3(p3) ] ,
 [ rE(:,planetpos3) ; vDep3(:,p3) ] , opts , mus ) ;
        [ t , stateT1r ] = ode45( @TwoBodyMotion , [ 0 dt1(r1) ] ,
 [ rE(:,planetpos1r) ; vDep1r(:,r1) ] , opts , mus ) ;
        [ t , stateT2r ] = ode45( @TwoBodyMotion , [ 0 dt2(r2) ] ,
 [ rE(:,planetpos2r) ; vDep2r(:,r2) ] , opts , mus ) ;
        [ t , stateT3r ] = ode45( @TwoBodyMotion , [ 0 dt3(r3) ] ,
 [ rE(:,planetpos3r) ; vDep3r(:,r3) ] , opts , mus ) ;
        for ii = 1:length(stateT1)
            rad1(ii) = norm( stateT1(ii,1:3) ) ;
        end
        for ii = 1:length(stateT2)
            rad2(ii) = norm( stateT2(ii,1:3) ) ;
        end
        for ii = 1:length(stateT3)
            rad3(ii) = norm( stateT3(ii,1:3) ) ;
        end
        for ii = 1:length(stateT1r)
            rad1r(ii) = norm( stateT1r(ii,1:3) ) ;
        end
        for ii = 1:length(stateT2r)
            rad2r(ii) = norm( stateT2r(ii,1:3) ) ;
        end
         for ii = 1:length(stateT3r)
            rad3r(ii) = norm( stateT3r(ii,1:3) ) ;
         end
        mrad(1) = min( rad1 ) ;
        mrad(2) = min( rad2 ) ;
        mrad(3) = min( rad3 ) ;
        mrad(4) = min( rad1r ) ;
        mrad(5) = min( rad2r ) ;
        mrad(6) = min( rad3r ) ;
% None go through the sun but the second retrograde orbit does
            disp([ 'The best departure time is ' ,
 num2str( timeDeparture(ceil(p2/24*k,1) )  ), ' hours after midnight
 on '])
            disp([ 'with a delta v of ' , num2str( dv_best2 ) , ' km/
s' ])
```

```
            disp([ 'arriving on 1 April 2019 at ' ,
num2str( timeArr2(ceil(p2/24*k,1) ) ) , ' hours' ])
            disp( 'This is a lot lower than without running the
different times' )

        [ tE , stateE ] = ode45( @TwoBodyMotion , [ 0 dt3(1) ].*4 ,
[ rE(:,1) ; vE(:,1) ] , opts , mus ) ;
        [ tV1 , stateV1 ] = ode45( @TwoBodyMotion , [ 0 dt1(1) ].*4 ,
[ rV1(:,1) ; vV1(:,1) ] , opts , mus ) ;
        [ tV2 , stateV2 ] = ode45( @TwoBodyMotion , [ 0 dt2(1) ].*4 ,
[ rV2(:,1) ; vV2(:,1) ] , opts , mus ) ;
        [ tV3 , stateV3 ] = ode45( @TwoBodyMotion , [ 0 dt3(1) ].*4 ,
[ rV3(:,1) ; vV3(:,1) ] , opts , mus ) ;


        figure
        hold on
        plot3( stateE(:,1) , stateE(:,2) , stateE(:,3) , 'k' )
        plot3( stateV1(:,1) , stateV1(:,2) , stateV1(:,3) )
        plot3( stateT1(:,1) , stateT1(:,2) , stateT1(:,3) , 'b' )
        plot3( stateT1r(:,1) , stateT1r(:,2) , stateT1r(:,3) , 'r' )
        title( 'First Arrival' )
        xlabel( 'X (km)' )
        ylabel( 'Y (km)' )
        zlabel( 'Z (km)' )
        legend( 'Earth' , 'Venus' , 'Transfer Prograde' , 'Transfer
Retrograde' )
        hold off

        figure
        hold on
        plot3( stateE(:,1) , stateE(:,2) , stateE(:,3) , 'k' )
        plot3( stateV2(:,1) , stateV2(:,2) , stateV2(:,3) )
        plot3( stateT2(:,1) , stateT2(:,2) , stateT2(:,3) , 'b' )
        plot3( stateT2r(:,1) , stateT2r(:,2) , stateT2r(:,3) , 'r' )
        title( 'Second Arrival' )
        xlabel( 'X (km)' )
        ylabel( 'Y (km)' )
        zlabel( 'Z (km)' )
        legend( 'Earth' , 'Venus' , 'Transfer Prograde' , 'Transfer
Retrograde' )
        hold off

        figure
        hold on
        plot3( stateE(:,1) , stateE(:,2) , stateE(:,3) , 'k' )
        plot3( stateV3(:,1) , stateV3(:,2) , stateV3(:,3) )
        plot3( stateT3(:,1) , stateT3(:,2) , stateT3(:,3) , 'b' )
        plot3( stateT3r(:,1) , stateT3r(:,2) , stateT3r(:,3) , 'r' )
        title( 'Third Arrival' )
        xlabel( 'X (km)' )
        ylabel( 'Y (km)' )
        zlabel( 'Z (km)' )
```

```
      legend( 'Earth' , 'Venus' , 'Transfer Prograde' , 'Transfer
Retrograde' )
      hold off
```

functions

```
   function dstate_dt = TwoBodyMotion( t , state , mu )
   % Finds change in state with respect to time. Input time, t, in
seconds and
   % state as position vector followed by velocity vector as well as
mu

   rad = norm( [ state(1) state(2) state(3) ] ) ; %radius

   dx = state(4) ; % velocity in x
   dy = state(5) ; % velocity in y
   dz = state(6) ; % velocity in z
   ddx = -mu*state(1)/rad^3 ; % acceleration in x
   ddy = -mu*state(2)/rad^3 ; % acceleration in y
   ddz = -mu*state(3)/rad^3 ; % acceleration in z

   dstate_dt = [ dx ; dy ; dz ; ddx ; ddy ; ddz ] ;

   end

       function [ v1 , v2 ] = Lamberts2( r1 , r2 , dt , mu , tol ,
pro )
   % pro is 1 or 0 for prograde or retrograde respectively

       r1mag = norm( r1 ) ;
       r2mag = norm( r2 ) ;
       rcross = cross( r1 , r2 ) ;

       % Find delta theta
           if pro == 1
               if rcross(3) >= 0
                   dtheta = acos( dot(r1,r2)/(r1mag*r2mag) ) ;
               else
                   dtheta = 2*pi - acos( dot(r1,r2)/(r1mag*r2mag) ) ;
               end
           else
               if rcross(3) < 0
                   dtheta = acos( dot(r1,r2)/(r1mag*r2mag) ) ;
               else
                   dtheta = 2*pi - acos( dot(r1,r2)/(r1mag*r2mag) ) ;
               end
           end

       A = sin( dtheta )*sqrt( r1mag*r2mag/(1-cos(dtheta)) ) ;
           z = 0 ;
           C = 1/2 ;
           S = 1/6 ;
           zup = 4*pi^2 ;
```

```
        zlow = -4*pi^2 ;
        y = r1mag + r2mag + (A*(z*S-1))/sqrt(C) ;
        chi = sqrt(y/C) ;
        dtloop = (chi^3*S)/sqrt(mu) + (A*sqrt(y))/sqrt(mu) ;
        while abs( dtloop - dt ) > tol
                if dtloop <= dt
                    zlow = z ;
                else
                    zup = z ;
                end
             z = ( zup + zlow ) / 2 ;
            [ S , C ] = Stumpf( z ) ;
            y = r1mag + r2mag + (A*(z*S-1))/sqrt(C) ;
            chi = sqrt(y/C) ;
            dtloop = (chi^3*S)/sqrt(mu) + (A*sqrt(y))/sqrt(mu) ;
        end
        f = 1 - y/r1mag ;
        g = A*sqrt(y/mu) ;
        gdot = 1 - y/r2mag ;

        v1 = ( 1/g )*( r2 - f*r1 ) ;
        v2 = ( 1/g )*( gdot*r2 - r1 ) ;
end
```

Published with MATLAB® R2018b