Contents

```
% MATLAB code for interplanetary transfer using one gravity assist
% maneuver. The following code does not account for planets phasing.
%
% Written by Marco Maggia, March 2016.

clc;clear;close all
```

%%%

```
                           EDITABLE SECTION                                 %
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% (1) Set altitude of parking orbit (circular)
hc_parking = 400; % [km]

% (2) Set Planets: Planet_1-->flyby planet, Planet_2-->target planet
Planet_1 = 'Jupiter';
Planet_2 = 'Uranus';

% (3) Set flyby side
side = 'Trailing';
% side = 'Leading';

% (4) Set altitude flyby at Planet_1
if strcmp(Planet_1,'Mars')
    h_flyby =  5000;
end
if strcmp(Planet_1,'Jupiter')
    h_flyby = 22400000;
end
if strcmp(Planet_1,'Saturn')
    h_flyby = 630000;
end
if strcmp(Planet_1,'Uranus')
    h_flyby = 240000;
end
if strcmp(Planet_1,'Neptune')
    h_flyby = 300000;
end

% (5) Set periapsis and apoapsis radius for target orbit around Planet_2
if strcmp(Planet_2,'Jupiter')
    rp_targ = 120000;
    ra_targ = 9000000;
end
if strcmp(Planet_2,'Saturn')
    rp_targ = 630000;
    ra_targ = 3*rp_targ;
end
if strcmp(Planet_2,'Uranus')
    rp_targ = 240000;
    ra_targ = 3*rp_targ;
end
if strcmp(Planet_2,'Neptune')
    rp_targ = 300000;
    ra_targ = 3*rp_targ;
end
if strcmp(Planet_2,'Pluto')
    rp_targ = 12500;
    ra_targ = 3*rp_targ;
end
```

%%%

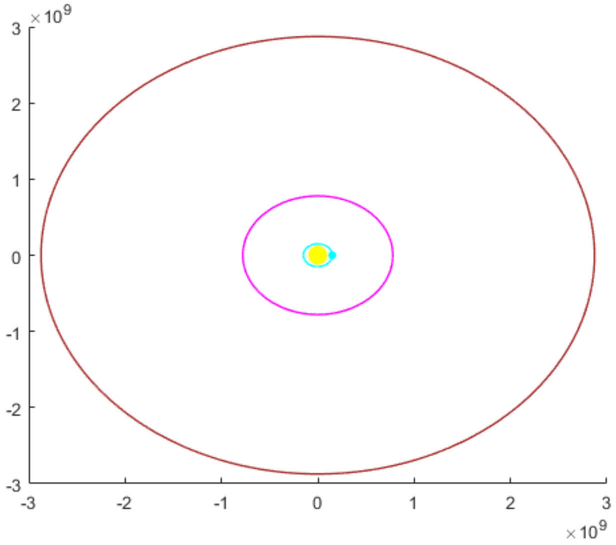
Parameters of Planets

```
[r_Earth,R_Earth,mu_Earth,T_Earth,n_Earth,v_Earth] = func_parameters_Planet('Earth');
[r_P1,R_P1,mu_P1,T_P1,n_P1,v_P1] = func_parameters_Planet(Planet_1);
[r_P2,R_P2,mu_P2,T_P2,n_P2,v_P2] = func_parameters_Planet(Planet_2);

% Target orbit
```

```
a_targ = 0.5*(ra_targ+rp_targ);
e_targ = (ra_targ-rp_targ)/(ra_targ+rp_targ);
h_targ = sqrt(mu_P2*rp_targ*(1+e_targ));
vp_targ = h_targ/rp_targ;
va_targ = h_targ/ra_targ;

if r_P1>=r_P2
   fprintf('Error! Combination of planets is invalid.\n\n')
end
```

Planets Orbits

```
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% @ March 5th 2016
% theta0_Earth   = 0;
% theta0_Mars    = 38.3699;
% theta0_Jupiter = 3.4097;
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~
options = odeset('reltol',1E-7,'reltol',1E-10);
mu_Sun = 1.3271244e11; %[km^3/s^2]

R0_Earth=eye(2);
R0_P1=eye(2);
R0_P2=eye(2);

p0_Earth = R0_Earth*[r_Earth,0]';
p0_P1    = R0_P1*[r_P1,0]';
p0_P2    = R0_P2*[r_P2,0]';

v0_Earth = R0_Earth*[0,v_Earth]';
v0_P1    = R0_P1*[0,v_P1]';
v0_P2    = R0_P2*[0,v_P2]';

X0_Earth = [p0_Earth',v0_Earth']';
X0_P1    = [p0_P1',v0_P1']';
X0_P2    = [p0_P2',v0_P2']';

t0 = 0;
[~,X_Earth] = ode45(@(t,X)func_ODE_R2BP(t,X,mu_Sun),[t0,T_Earth],X0_Earth,options);
[~,X_P1]    = ode45(@(t,X)func_ODE_R2BP(t,X,mu_Sun),[t0,T_P1]   ,X0_P1,options);
[~,X_P2]    = ode45(@(t,X)func_ODE_R2BP(t,X,mu_Sun),[t0,T_P2]   ,X0_P2,options);


figure(1),hold on
plot(0,0,'marker','o','markersize',10.0,'markeredgecolor','y','markerfacecolor','y')
plot(X_Earth(:,1)   ,X_Earth(:,2)   ,'linewidth',1.0,'linestyle','-','color','c')
plot(X_P1(:,1),X_P1(:,2)    ,'linewidth',1.0,'linestyle','-','color','m')
plot(X_P2(:,1),X_P2(:,2) ,'linewidth',1.0,'linestyle','-','color',[165/255,42/255,42/255])
plot(X_Earth(1,1)   ,X_Earth(1,2)   ,'marker','o','markersize',4.0,'markeredgecolor','c','markerfacecolor','c')
```



Hohmann Earth-Planet1 and Earth-Planet2

```
v_inf_dep_Hohmann_EP1 = norm(sqrt(2*mu_Sun)*sqrt(r_P1/(r_Earth*(r_Earth+r_P1))) - v_Earth);
v_inf_dep_Hohmann_EP2 = norm(sqrt(2*mu_Sun)*sqrt(r_P2/(r_Earth*(r_Earth+r_P2))) - v_Earth);
v_inf_arr_Hohmann_EP2 = norm(sqrt(2*mu_Sun)*sqrt(r_Earth/(r_P2*(r_Earth+r_P2))) - v_P2);

r_parking_dep = hc_parking + R_Earth;
v_parking_dep = sqrt(mu_Earth/r_parking_dep);

a_EP1         = (r_Earth+r_P1)/2;
a_EP2         = (r_Earth+r_P2)/2;
```

```
T_Hohmann_EP1 = pi/sqrt(mu_Sun)*a_EP1^1.5;
T_Hohmann_EP2 = pi/sqrt(mu_Sun)*a_EP2^1.5;

v_per_hyp_dep_Hohmann = sqrt(v_inf_dep_Hohmann_EP2^2 + 2*mu_Earth/r_parking_dep);
DV_dep_Hohmann        = norm(v_per_hyp_dep_Hohmann - v_parking_dep);

v_per_hyp_arr_Hohmann = sqrt(v_inf_arr_Hohmann_EP2^2 + 2*mu_P2/rp_targ);
DV_arr_Hohmann        = norm(vp_targ - v_per_hyp_arr_Hohmann);

DV_tot_Hohmann = DV_dep_Hohmann+DV_arr_Hohmann;

v_inf_dep_Hohmann = [0,v_inf_dep_Hohmann_EP2]';
p0_sc_Hohmann = [r_Earth, 0]';
v0_sc_Hohmann = [0,v_Earth]' + v_inf_dep_Hohmann;
X0_sc_Hohmann = [p0_sc_Hohmann',v0_sc_Hohmann']';
[~,X_sc_Hohmann]  = ode45(@(t,p)func_ODE_R2BP(t,p,mu_Sun),[t0,T_Hohmann_EP2],X0_sc_Hohmann,options);

flag_stop = 0;
% You can modify v_max, N, dv_inf
v_max = v_inf_dep_Hohmann_EP2 + 1.0;
i_max=100;
dv_inf = 0.02; %[km/s]

i=1;
while flag_stop == 0 && i<i_max
```

```
    % v_inf at departure
    v_inf_dep(:,i) = [0, v_max - (i-1)*dv_inf]';
%     v_inf_dep(:,i) = [0, v_max - dv_inf*i]';

    % Initial conditions for 1st transfer
    p0_sc1(:,i) = [r_Earth, 0];                   % position vector
    v0_sc1(:,i) = [0,v_Earth]' + v_inf_dep(:,i);  % velocity vector
    X0_sc1(:,i) = [p0_sc1(:,i)',v0_sc1(:,i)']';   % position+velocity vector

    h_sc1(:,i)      = cross([p0_sc1(:,i)',0]',[v0_sc1(:,i)',0]');  % Angular momentum of orbit segment 1
    norm_h_sc1(i,1) = norm(h_sc1(:,i));
    e_sc1(:,i) = cross([v0_sc1(:,i)',0]',h_sc1(:,i))/mu_Sun - [p0_sc1(:,i)',0]'/norm([p0_sc1(:,i)',0]');
    norm_e_sc1(i,1) = norm(e_sc1(:,i));

    % func_event_sc stops the integration when the sc intersect the orbit
    % of the first planet
    options_sc = odeset('Events',@(t,X)func_event_sc(r_P1,t,X),'RelTol',1E-7,'AbsTol',1E-10);
    tf = T_Hohmann_EP1;

    % Integration of first transfer from Earth to Planet 1
    [t_sc1,X_sc1]  = ode45(@(t,p)func_ODE_R2BP(t,p,mu_Sun),[t0,tf],X0_sc1(:,i),options_sc);

    theta_P1_arr(i,1) = atan2d(X_sc1(end,2),X_sc1(end,1));
    theta_P1_dep(i,1) = theta_P1_arr(i,1) - 360/T_P1*t_sc1(end);

    theta_E_arr(i,1)  = 360/T_Earth*t_sc1(end);
    phi_E_P1(i,1) = theta_P1_dep(i,1);


    % radius of s/c on first transfer orbit segment
    r_sc1 = sqrt(X_sc1(:,1).^2 + X_sc1(:,2).^2);
```

Discard the cases in which the s/c doesn't reach the first planet

```
    if find(r_sc1>=r_P1)~=0
```

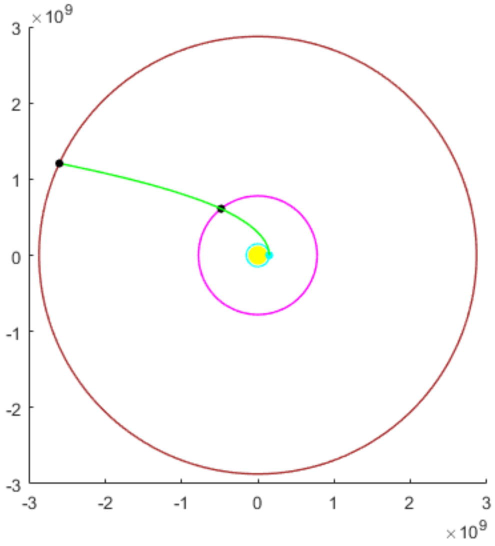
```
        % time at flyby
        t_flyby(i,1) = t_sc1(end);

        % position and velocity vectors at flyby (at the entrance of the SOI of
        % Planet 1)
        p0_sc2(:,i) = [X_sc1(end,1),X_sc1(end,2)];
        v1_fb(:,i)  = [X_sc1(end,3),X_sc1(end,4)]; % fb: flyby
        norm_v1_fb(i,1)=norm(v1_fb(:,i));

        % position and velocity vectors at flyby of Planet 1
        p_P1_fb(:,i)  = p0_sc2(:,i);
        v_P1_fb(:,i)  = [cosd(90)  -sind(90); sind(90)  cosd(90)]*p_P1_fb(:,i)/norm(p_P1_fb(:,i))*v_P1;

        % v_inf at entrance of SOI of Planet 1
        v_inf_fb_1(:,i) = v1_fb(:,i)-v_P1_fb(:,i);

        % Flight path angle and PHI_1 before flyby
        gamma_1(i,1) = acosd((v_P1_fb(:,i)'*v1_fb(:,i)     )/(norm(v_P1_fb(:,i))*norm(v1_fb(:,i))));
        phi_1(i,1)   = acosd((v_P1_fb(:,i)'*v_inf_fb_1(:,i))/(norm(v_P1_fb(:,i))*norm(v_inf_fb_1(:,i))));

        % Parameters at Flyby
        r_flyby = R_P1+h_flyby;   % Flyby radius
        beta_arr(i,1) = acosd(1/(1 + r_flyby*norm(v_inf_fb_1(:,i))^2/mu_P1));

        if strcmp(side,'Trailing')
        delta_arr(i,1) = (180 - 2*beta_arr(i,1));
    else
        delta_arr(i,1) = -(180 - 2*beta_arr(i,1));
    end

    % Rotation matrix for rotating v_inf
    R1_2 = [cosd(delta_arr(i,1))  -sind(delta_arr(i,1))
            sind(delta_arr(i,1))   cosd(delta_arr(i,1))];

    % v_inf at the exit of SOI of Planet 1
    v_inf_fb_2(:,i) = R1_2*v_inf_fb_1(:,i);

    % velocity of s/c after flyby
    v2_fb(:,i) = v_P1_fb(:,i) + v_inf_fb_2(:,i);
    norm_v2_fb(i,1)=norm(v2_fb(:,i));

    % Flight path angle and PHI_1 before flyby
    gamma_2(i,1) = acosd((v_P1_fb(:,i)'*v2_fb(:,i)     )/(norm(v_P1_fb(:,i))*norm(v2_fb(:,i))));
    phi_2(i,1)   = acosd((v_P1_fb(:,i)'*v_inf_fb_2(:,i))/(norm(v_P1_fb(:,i))*norm(v_inf_fb_2(:,i))));

    % Initial conditions for 2nd transfer
    v0_sc2(:,i) = v2_fb(:,i);
    X0_sc2(:,i) = [p0_sc2(:,i)',v0_sc2(:,i)']';

    h_sc2(:,i)      = cross([p0_sc2(:,i)',0]',[v0_sc2(:,i)',0]');  % Angular momentum of orbit segment 2
    norm_h_sc2(i,1) = norm(h_sc2(:,i));
    e_sc2(:,i) = cross([v0_sc2(:,i)',0]',h_sc2(:,i))/mu_Sun - [p0_sc2(:,i)',0]'/norm([p0_sc2(:,i)',0]');
    norm_e_sc2(i,1) = norm(e_sc2(:,i));

    % Integrating second segment of transfer
    tf = T_Hohmann_EP2;
    options_sc = odeset('Events',@(t,X)func_event_sc(r_P2,t,X),'RelTol',1E-7,'AbsTol',1E-10);
    [t_sc2,X_sc2]  = ode45(@(t,p)func_ODE_R2BP(t,p,mu_Sun),[t0,tf],X0_sc2(:,i),options_sc);

    % Radius of s/c orbit on second segment
    r_sc2 = sqrt(X_sc2(:,1).^2 + X_sc2(:,2).^2);
```

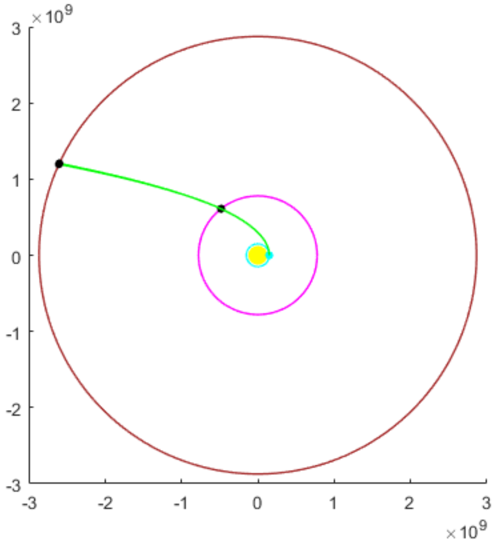
Plots in grey the trajectories that fail to reach Planet 2

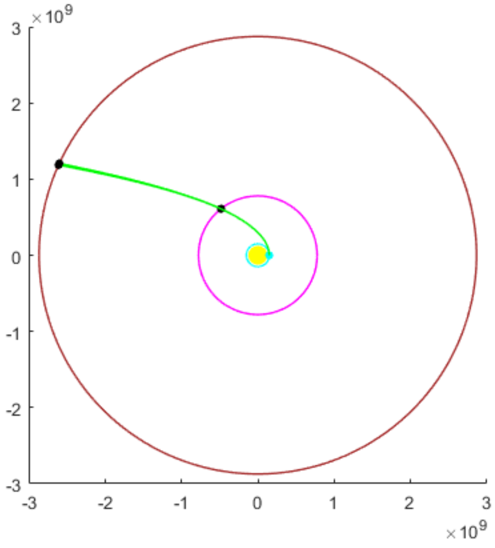
Plots in colors (blue to green) the trajectories that reach Planet 2

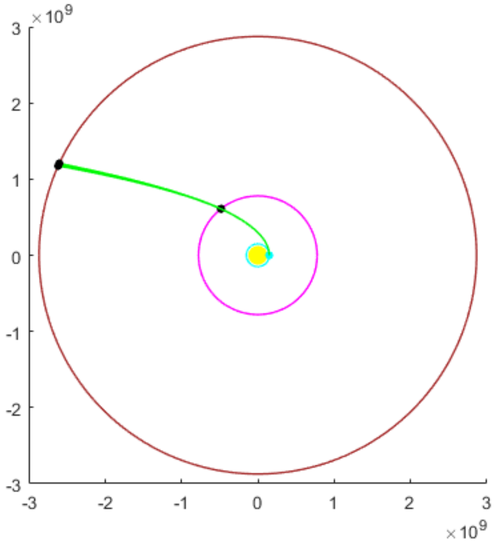
```
    if find(r_sc2>=r_P2)~=0
```

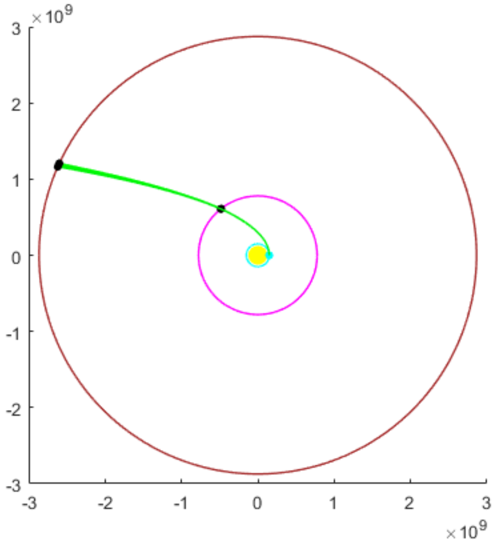
```
        % colors for plotsfunc_ODE
        traj_color = [0,(i_max+1-i)/i_max,(i-1)/i_max];
        figure(1),hold on
        plot(p0_sc2(1,i),p0_sc2(2,i),'marker','o','markersize',4.0,'markeredgecolor','k','markerfacecolor','k')
        plot(X_sc1(:,1),X_sc1(:,2)    ,'linewidth',1.0,'linestyle','-','color',traj_color)
        plot(X_sc2(:,1),X_sc2(:,2)    ,'linewidth',1.0,'linestyle','-','color',traj_color)
        plot(X_sc2(end,1),X_sc2(end,2),'marker','o','markersize',4.0,'markeredgecolor','k','markerfacecolor','k')
        daspect([1,1,1])

        Transfer_Time1(i,1) = t_sc1(end);
        Transfer_Time2(i,1) = t_sc2(end);
        Total_Time(i,1) = t_sc1(end) + t_sc2(end);

        % position and velocity vectors at flyby (at the entrance of the SOI of
        % Planet 2)
        p0_sc3(:,i) = [X_sc2(end,1),X_sc2(end,2)];
        v3_fb(:,i)  = [X_sc2(end,3),X_sc2(end,4)]; % fb: flyby

        % position and velocity vectors at flyby of Planet 2
        p_P2_fb(:,i)  = p0_sc3(:,i);
        v_P2_fb(:,i)  = [cosd(90)  -sind(90); sind(90)  cosd(90)]*p_P2_fb(:,i)/norm(p_P2_fb(:,i))*v_P2;
```

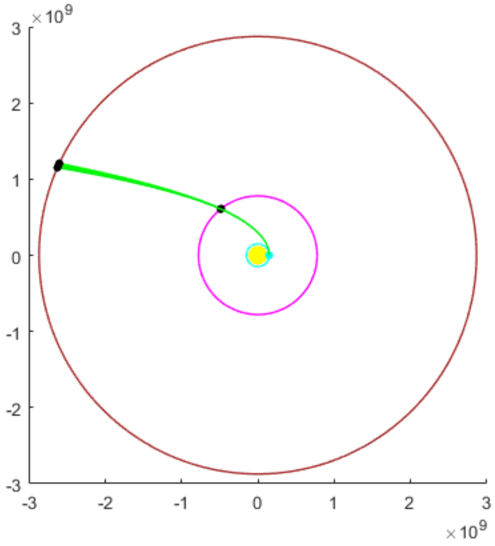

×10^9
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
×10^9

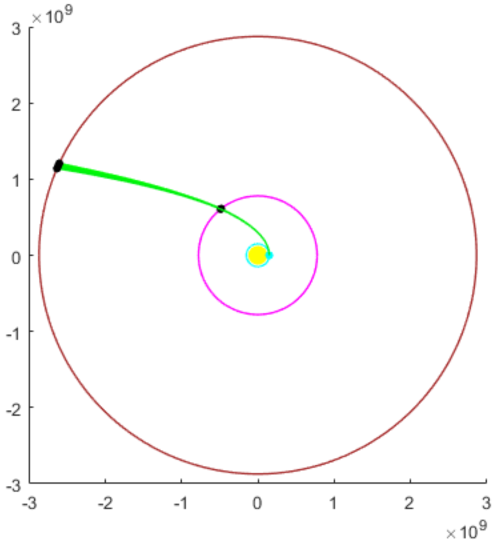

×10^9
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
×10^9


×10^9
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
×10^9

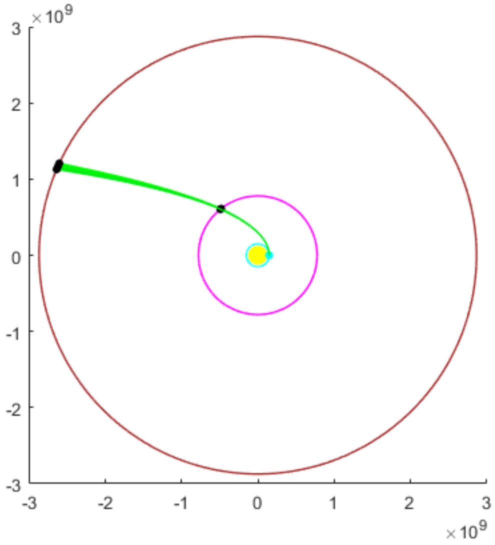

×10^9
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
×10^9


×10^9
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
×10^9

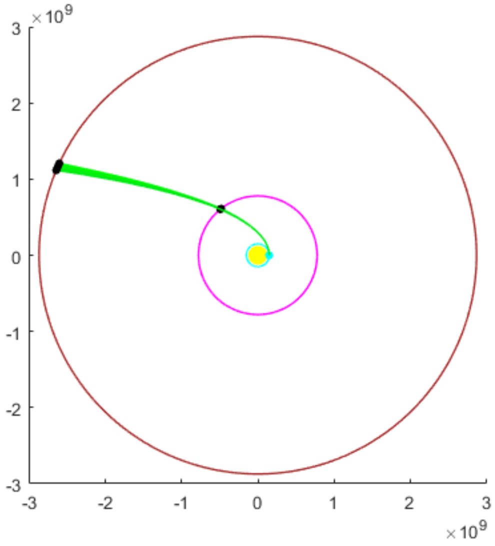

×10^9
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
×10^9



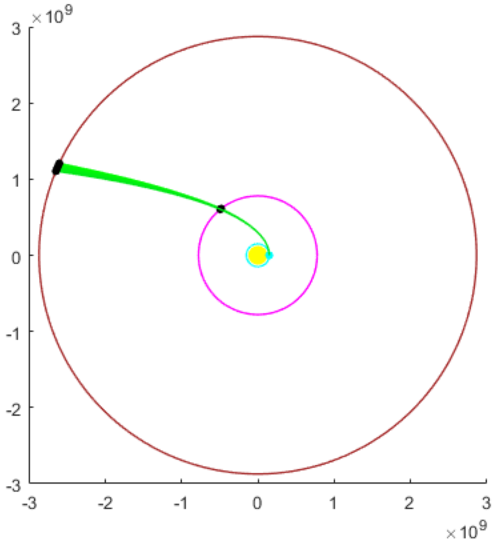
3
2
1
0
-1
-2
-3
× 10^9
-3
-2
-1
0
1
2
3
× 10^9

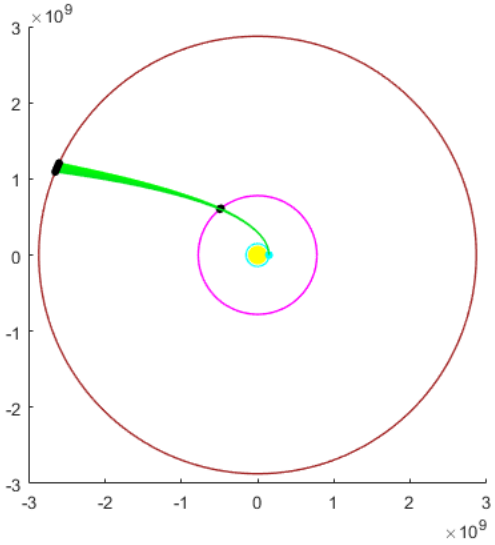


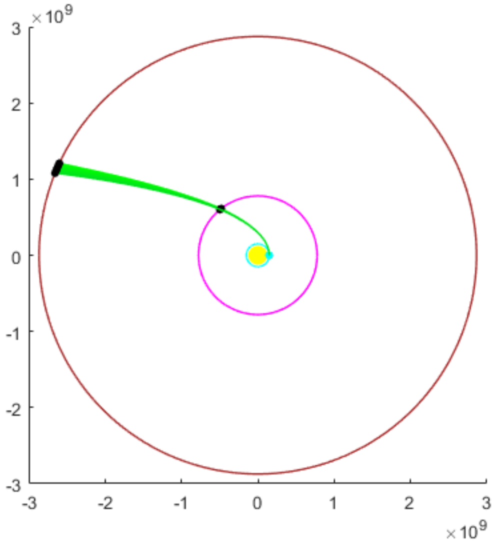
3
2
1
0
-1
-2
-3
× 10^9
-3
-2
-1
0
1
2
3
× 10^9

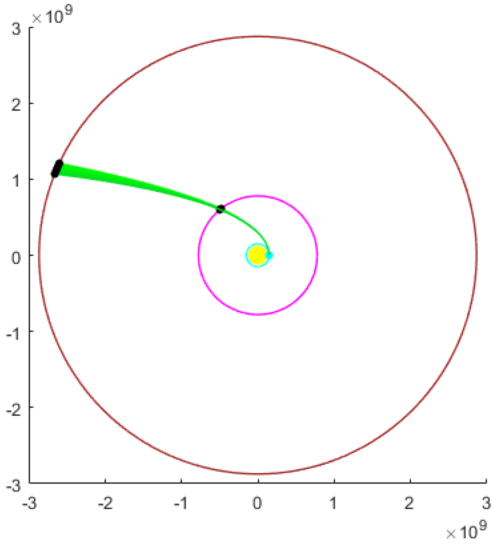


3
2
1
0
-1
-2
-3
× 10^9
-3
-2
-1
0
1
2
3
× 10^9

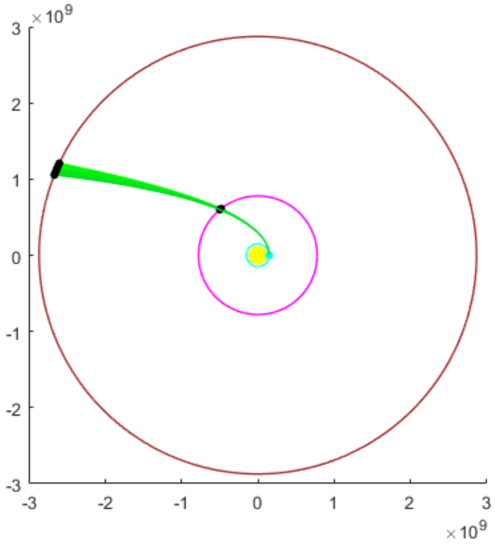

×10^9
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
×10^9


×10^9
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
×10^9

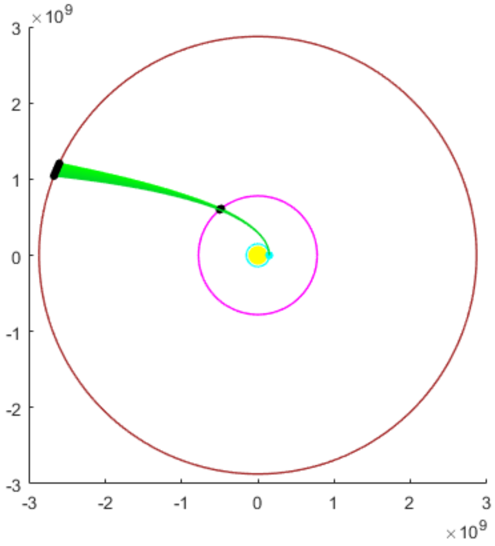

×10^9
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
×10^9



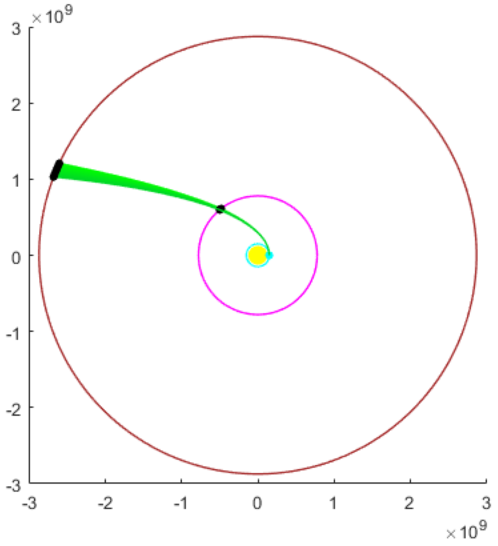
×10^9
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
×10^9

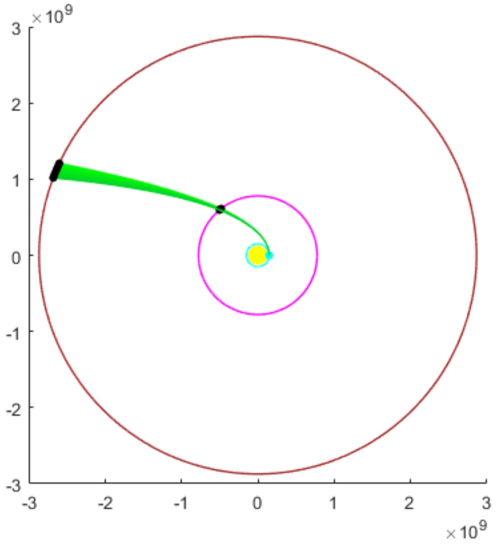


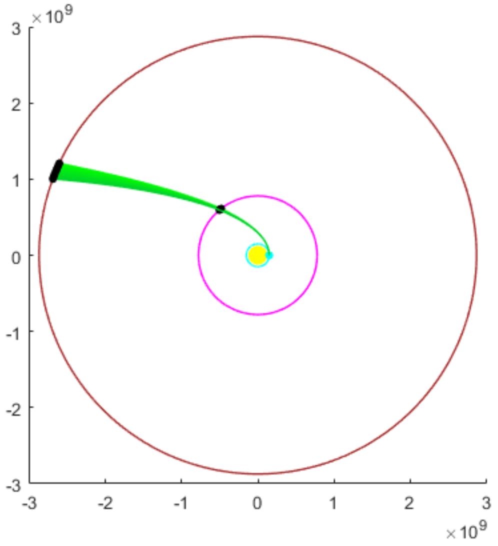
×10^9
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
×10^9

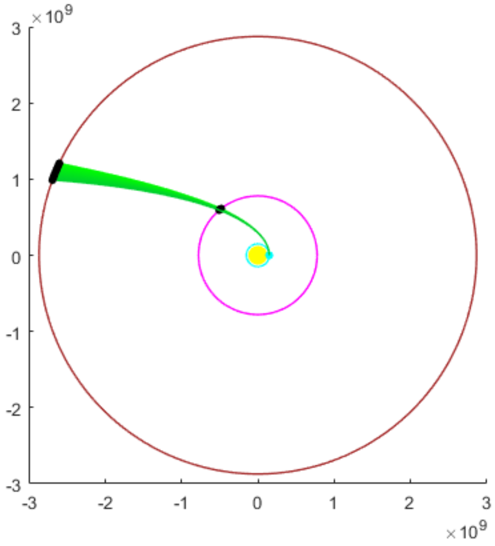


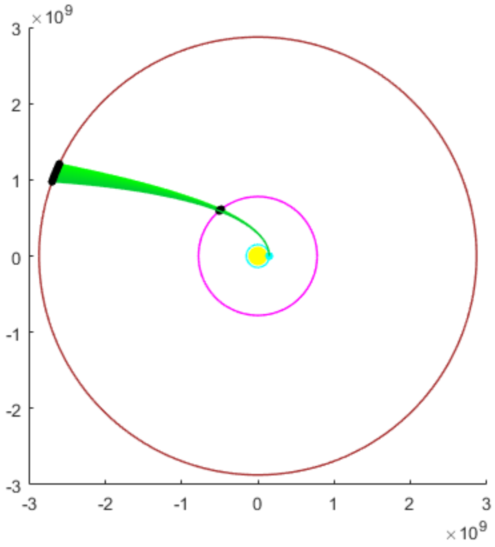
×10^9
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
×10^9

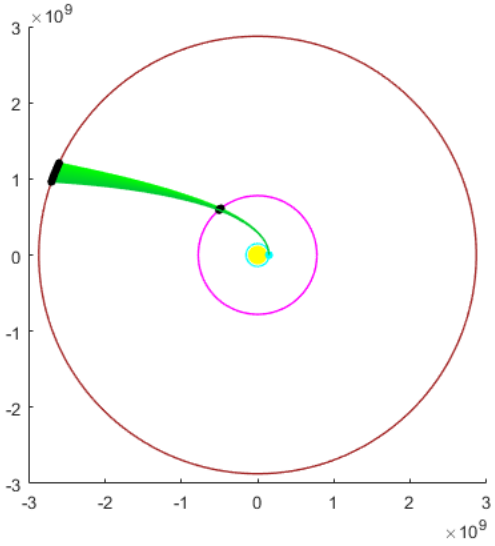

×10^9
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
×10^9

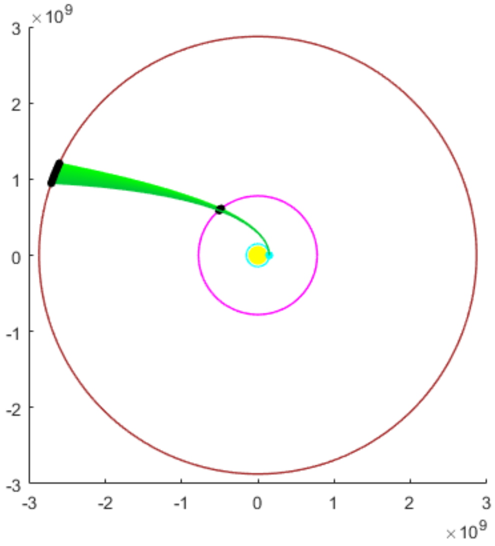

×10^9
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
×10^9

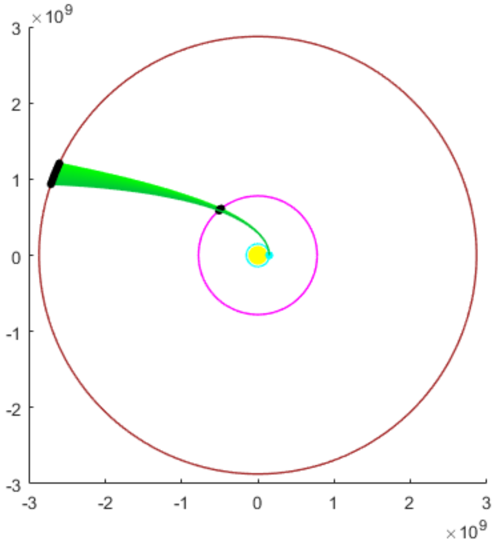

×10^9
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
×10^9

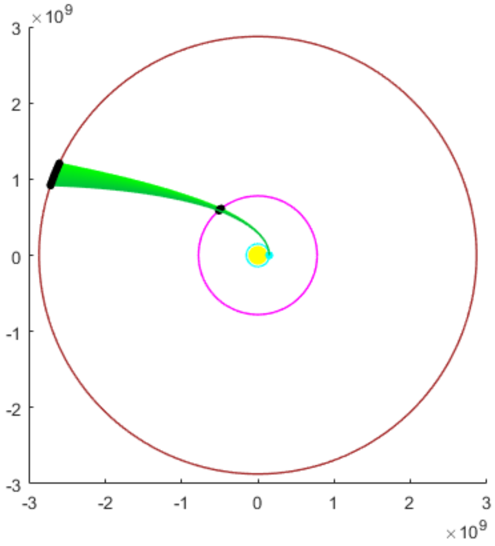

× 10^9
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
× 10^9

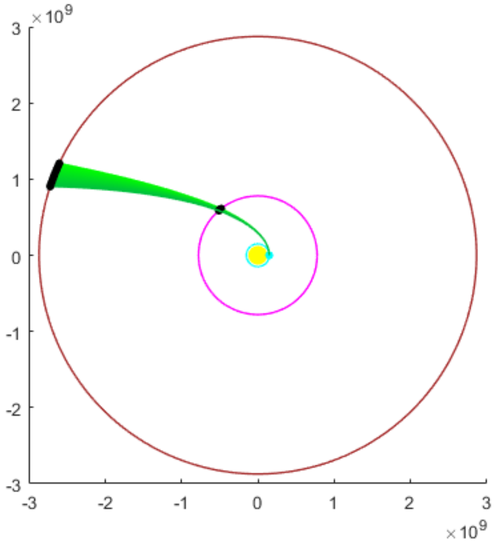

× 10^9
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
× 10^9

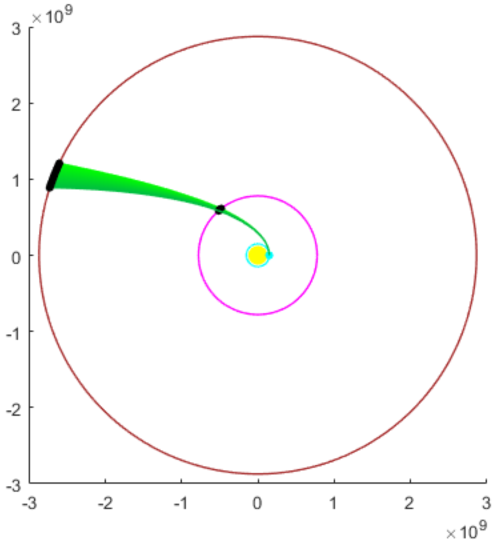

× 10^9
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
× 10^9

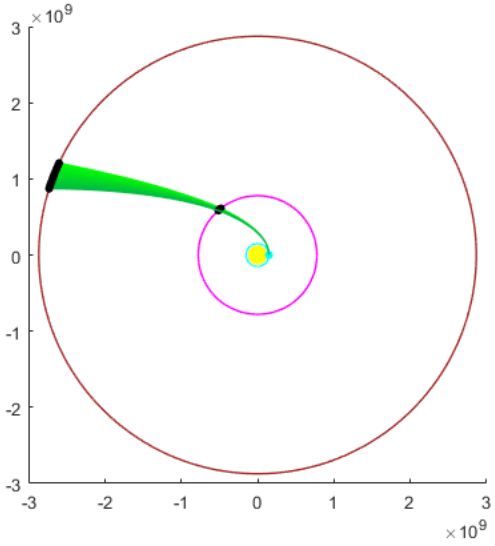

× 10^9
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
× 10^9

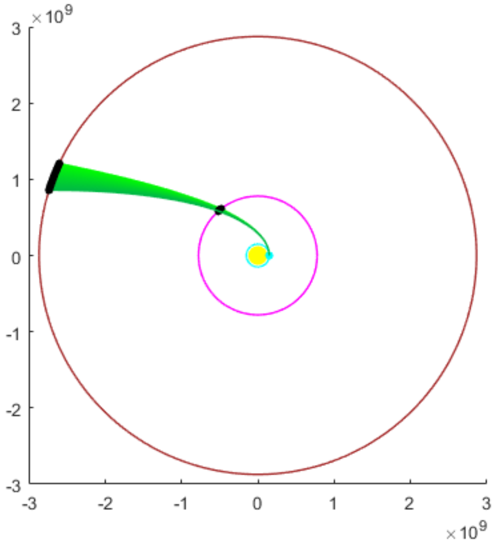

× 10^9
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
× 10^9


× 10^9
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
× 10^9

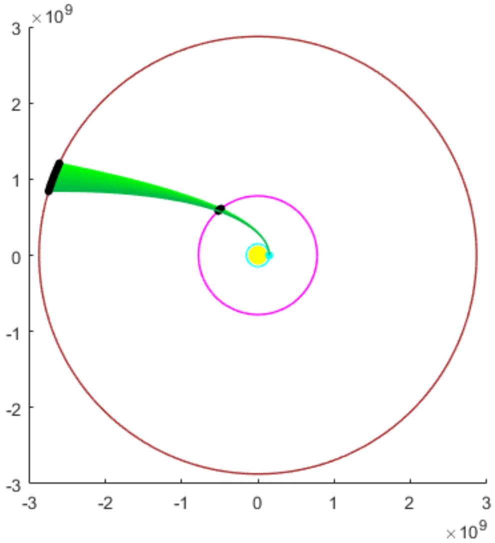

×10^9
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
×10^9


×10^9
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
×10^9

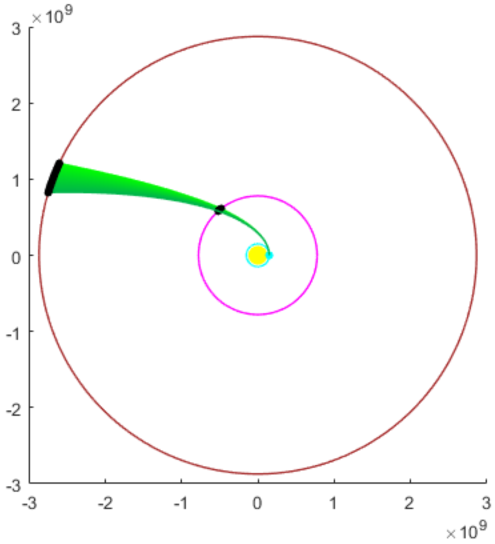

×10^9
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
×10^9



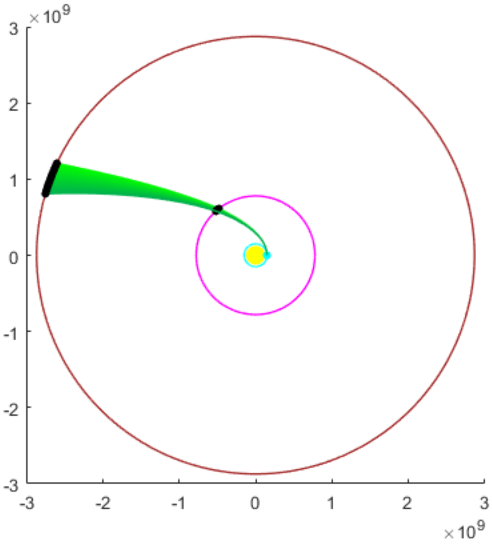
×10^9
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
×10^9



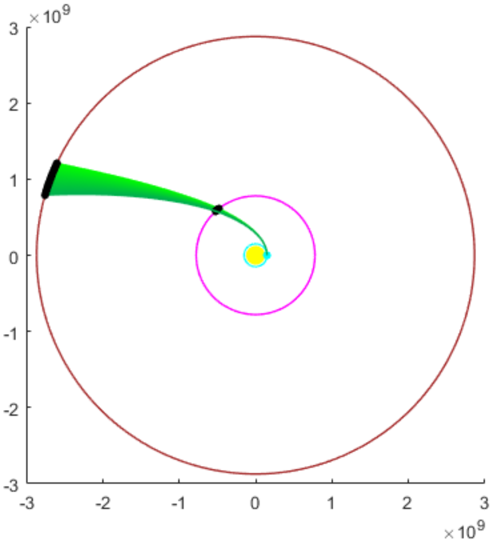
×10^9
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
×10^9



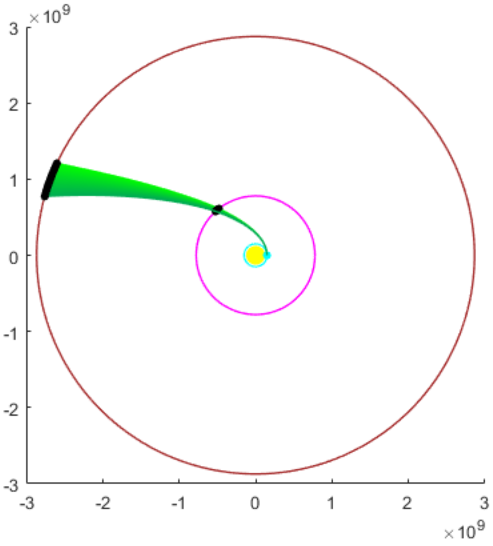
×10^9
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
×10^9



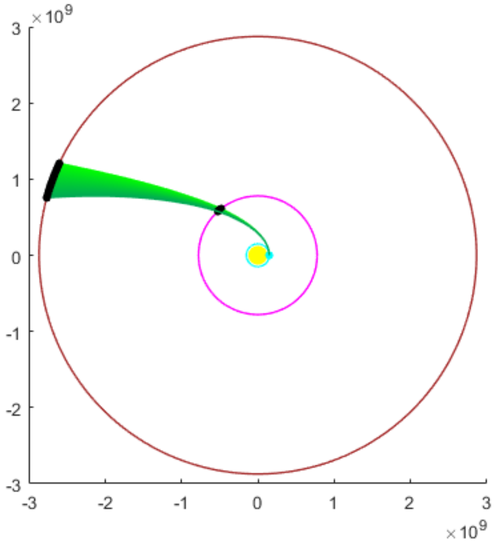
×10^9
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
×10^9



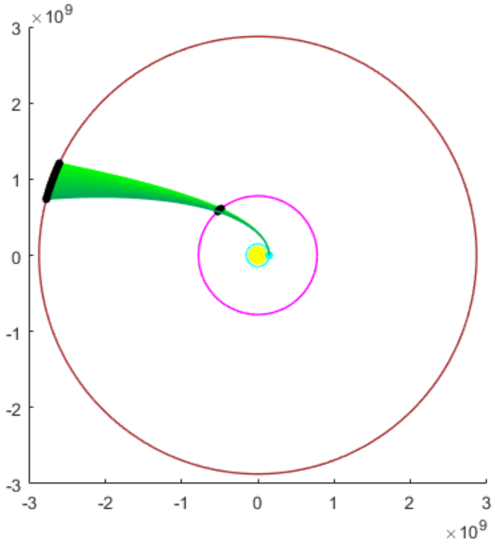
×10^9
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
×10^9



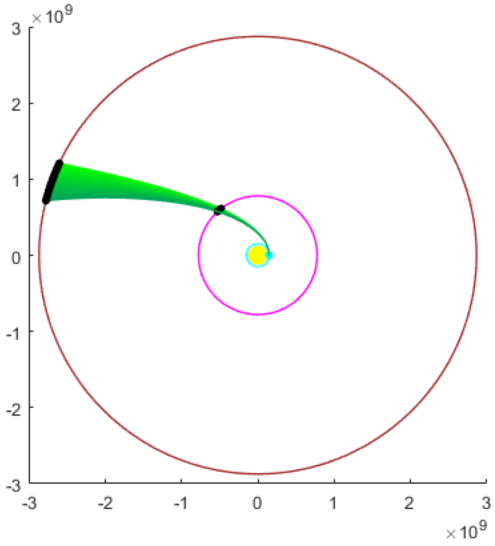
×10^9
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
×10^9



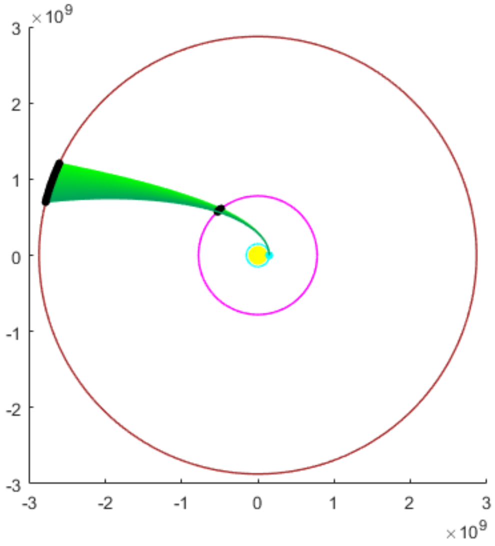
× 10^9
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
× 10^9

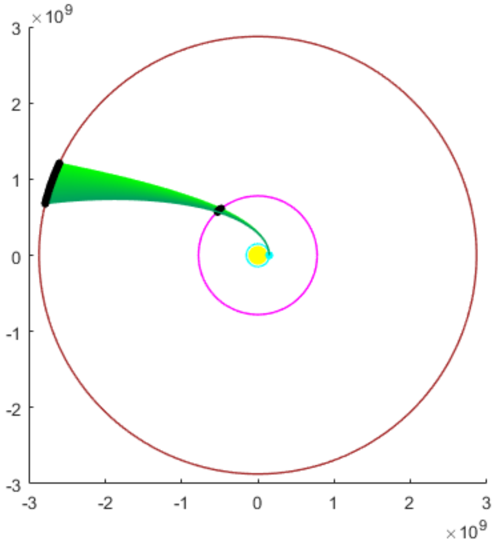


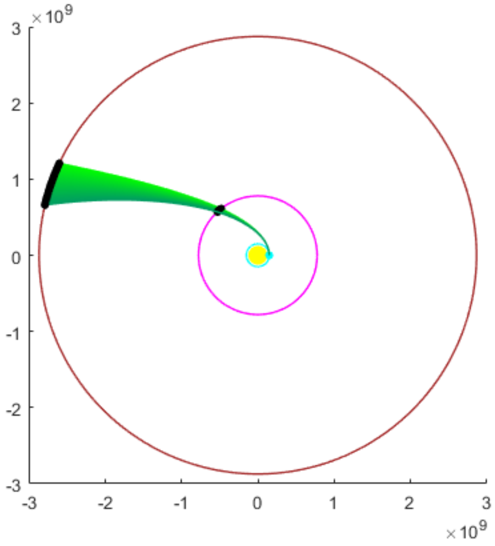
× 10^9
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
× 10^9

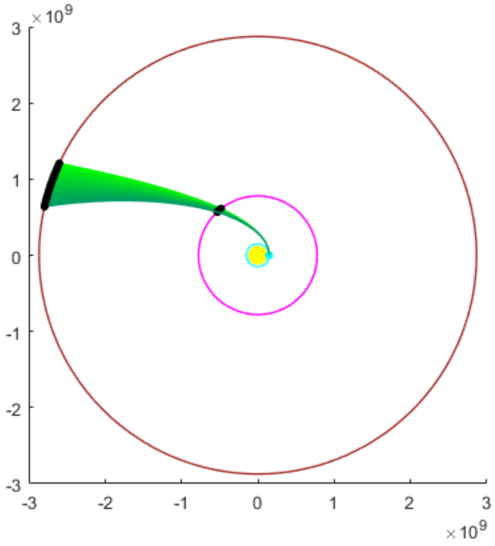


× 10^9
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
× 10^9

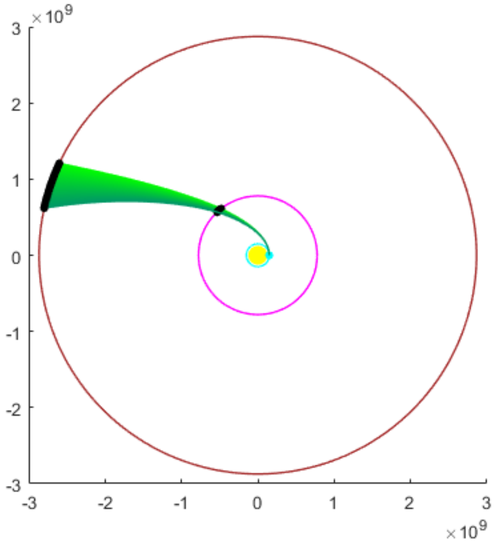

×10^9
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
×10^9


×10^9
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
×10^9

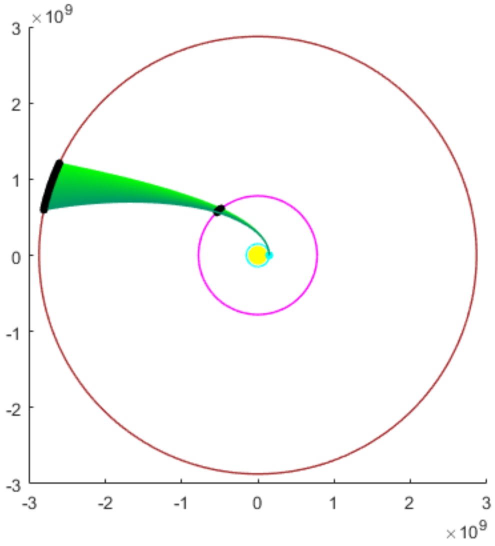

×10^9
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
×10^9



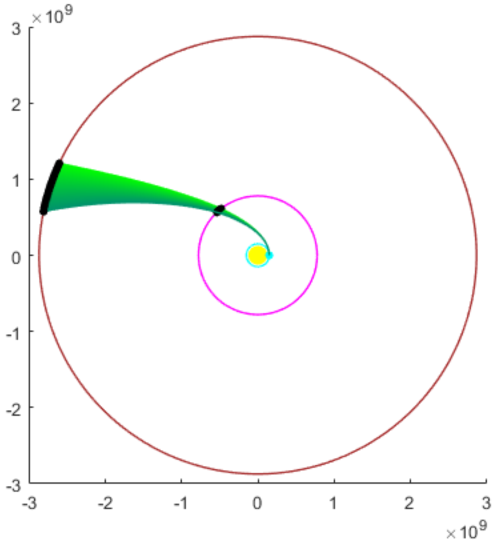
× 10^9
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
× 10^9

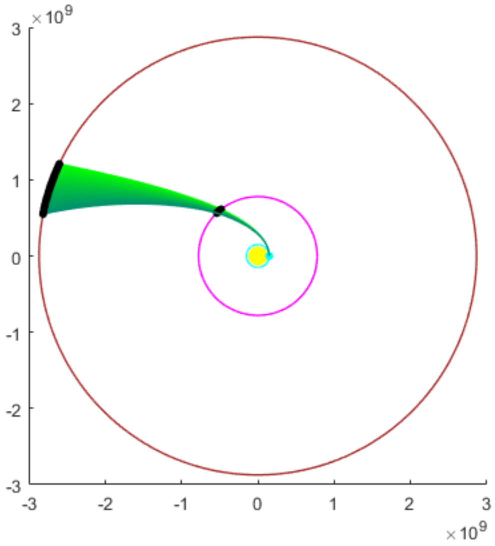


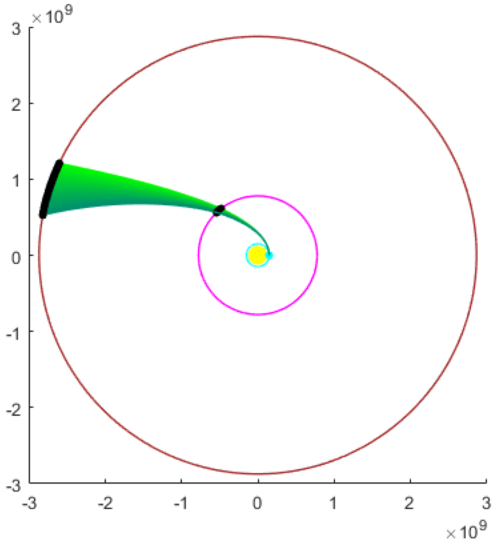
× 10^9
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
× 10^9

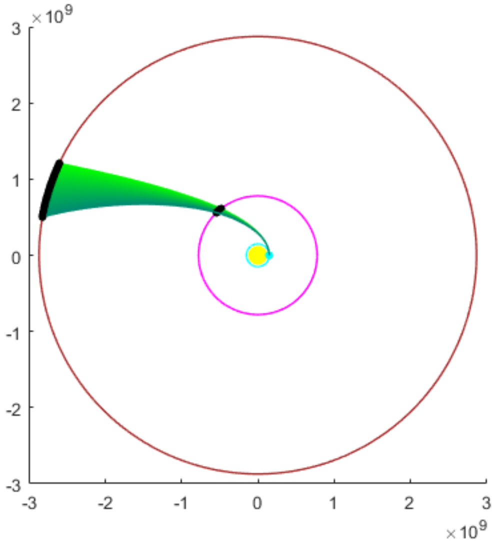


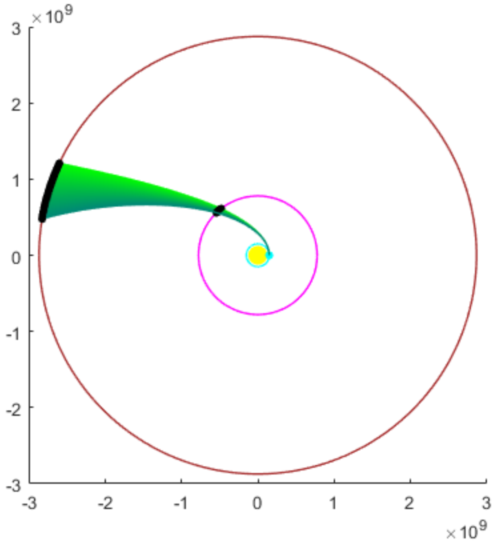
× 10^9
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
× 10^9

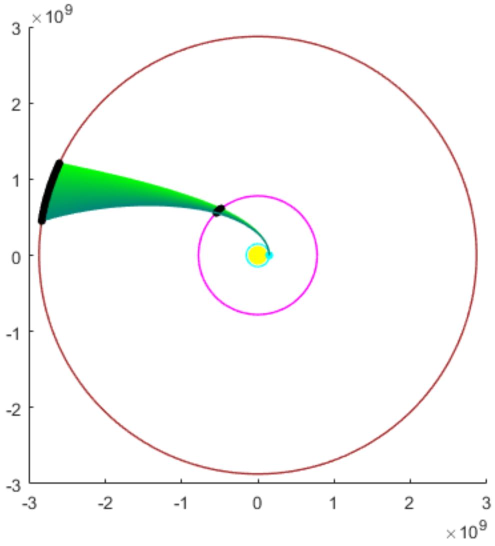

×10^9
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
×10^9

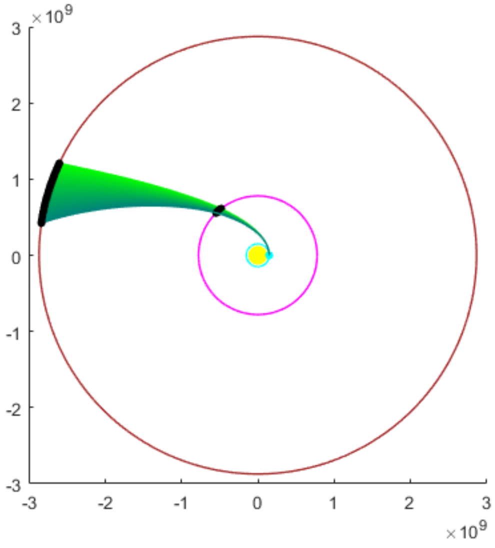

×10^9
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
×10^9

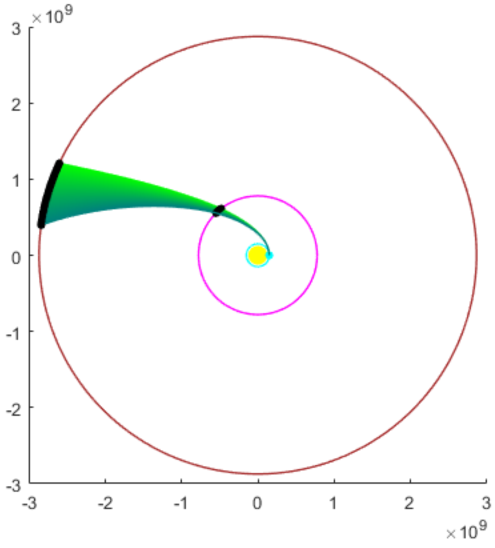

×10^9
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
×10^9

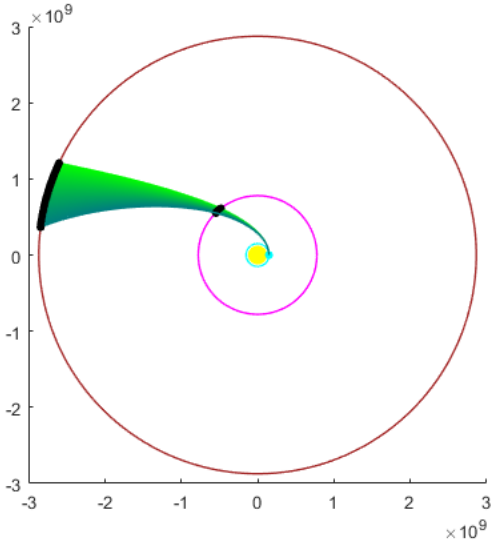

× 10^9
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
× 10^9

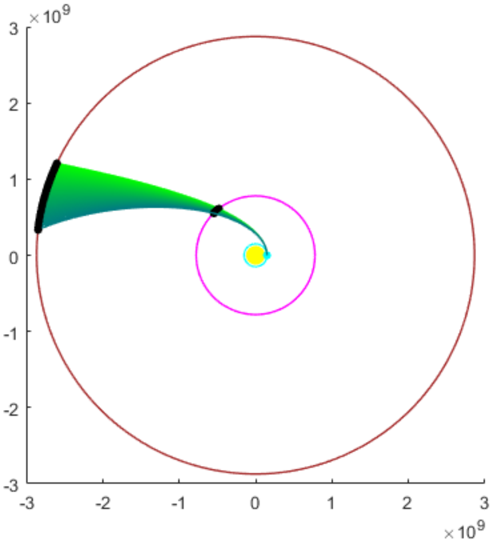

× 10^9
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
× 10^9

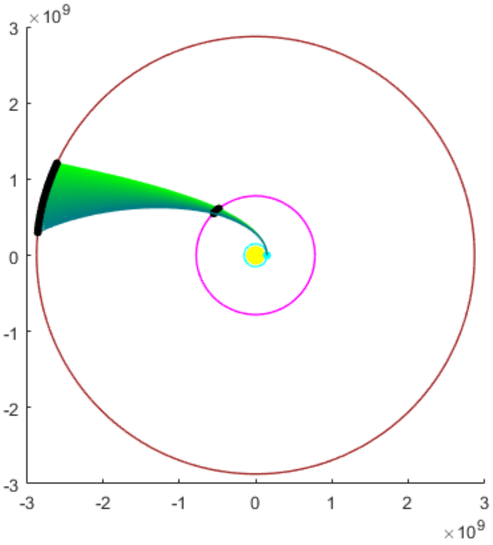

× 10^9
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
× 10^9

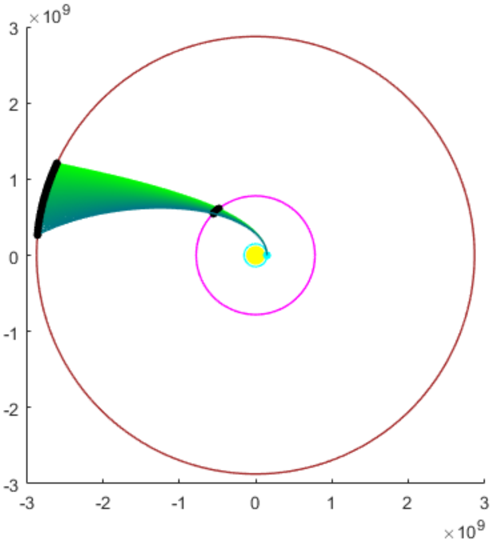

×10^9
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
×10^9

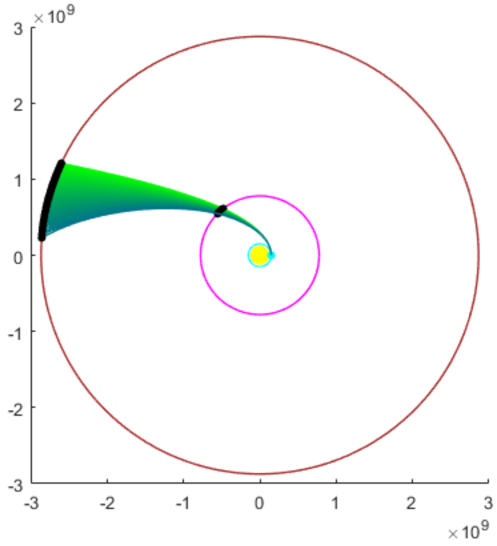

×10^9
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
×10^9

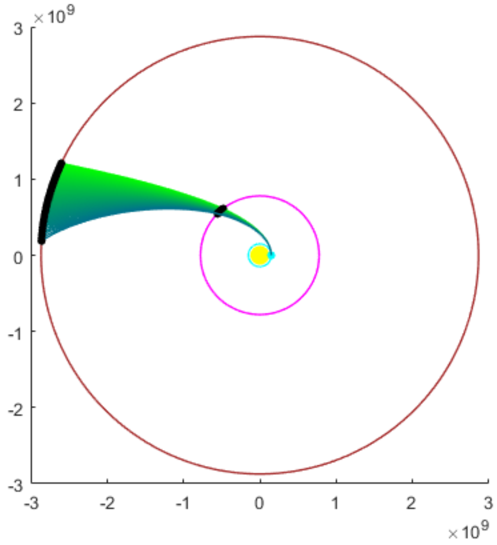

×10^9
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
×10^9

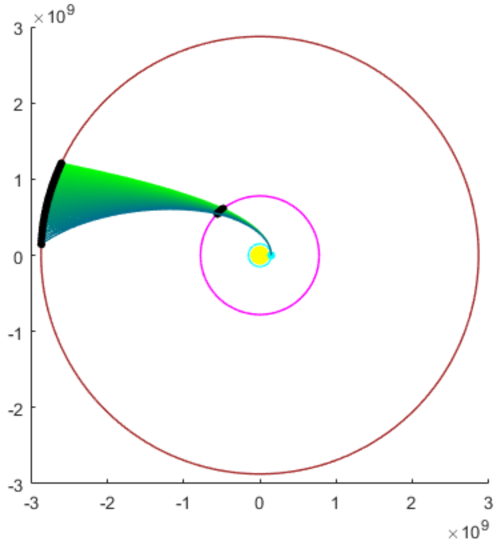

×10^9
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
×10^9

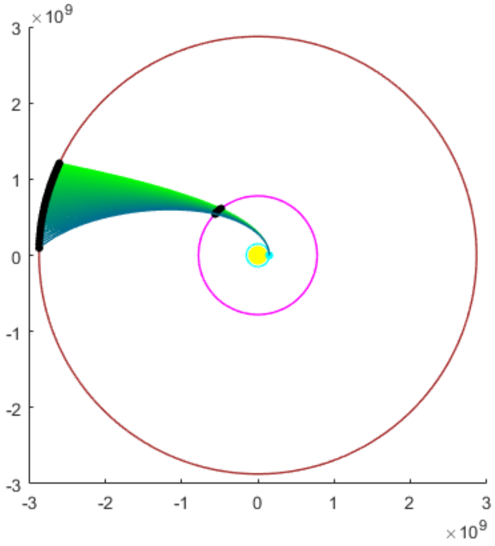

×10^9
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
×10^9

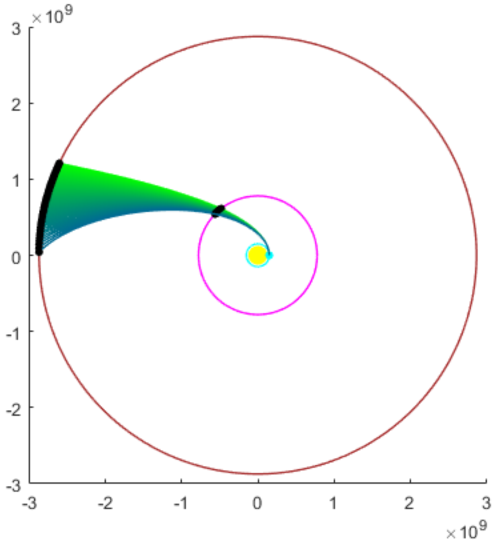

×10^9
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
×10^9

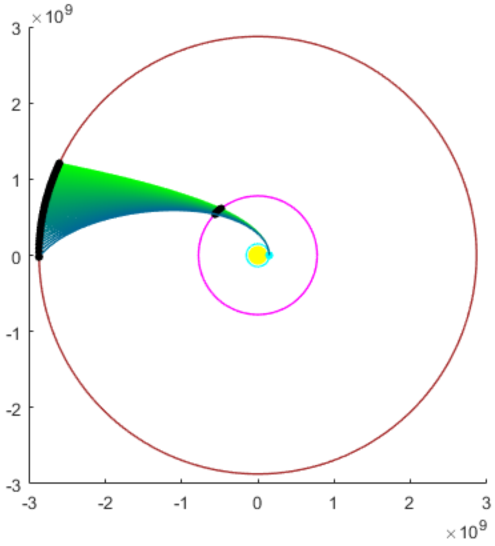

×10^9
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
×10^9

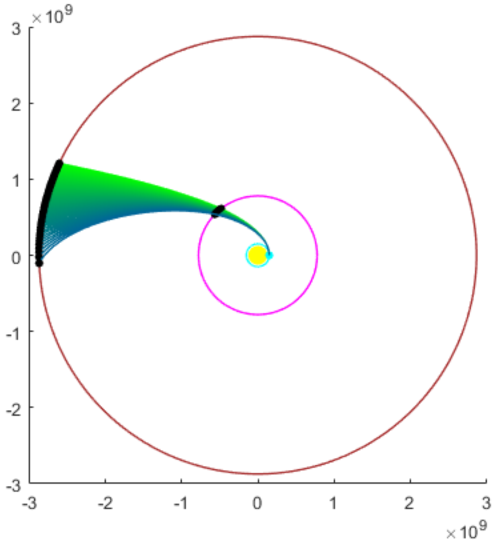

×10^9
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
×10^9

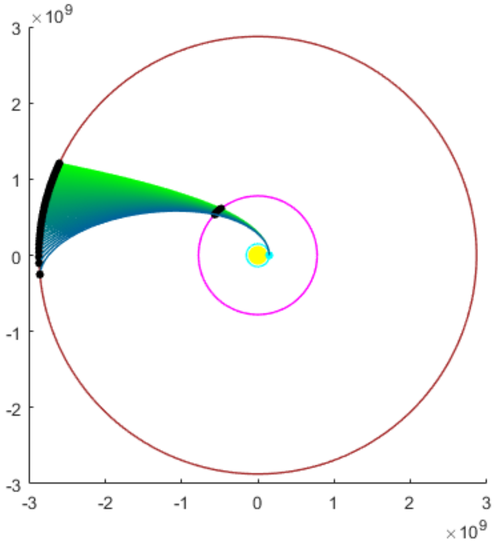

×10^9
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
×10^9


×10^9
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
×10^9


×10^9
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
×10^9


×10^9
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
×10^9





Insertion into hyperbolic orbit

norm of v_inf_dep

```
            norm_v_inf_dep(i,1) = norm(v_inf_dep(:,i));
            % norm of v hyperbolic at perigeedeparture
            v_per_hyp_dep(i,1) = sqrt(norm_v_inf_dep(i,1)^2 + 2*mu_Earth/r_parking_dep);
            % DV for hyperbolic orbit insertion
            DV_dep(i,1) = v_per_hyp_dep(i,1) - v_parking_dep;
```

Insertion into target orbit

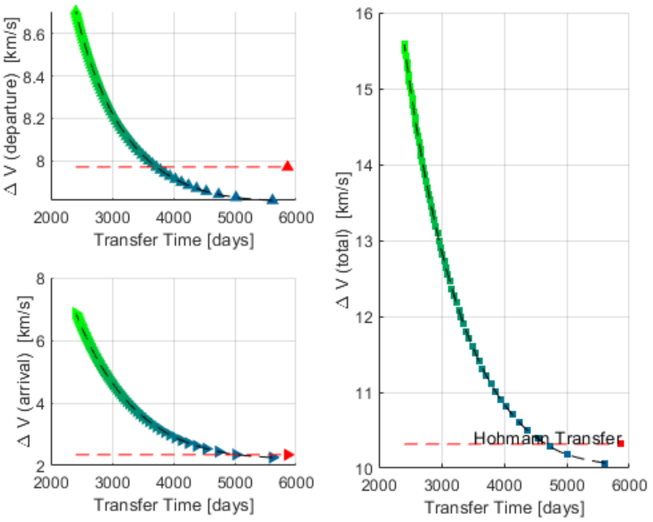
v_inf at entrance of SOI of Planet 2

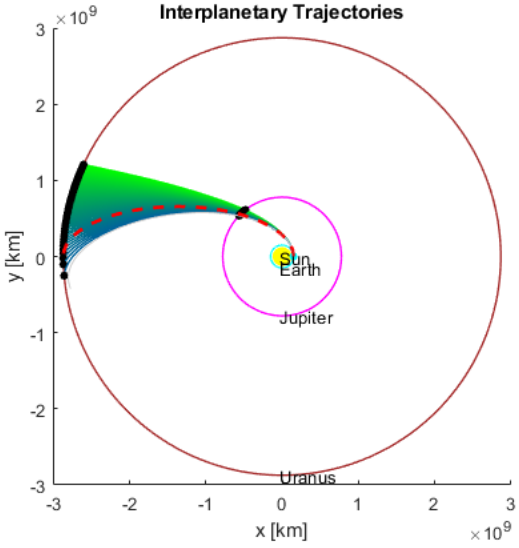
```
            v_inf_fb_3(:,i) = v3_fb(:,i)-v_P2_fb(:,i);
            % norm of v_inf
            v_inf_arr(i,1) = norm(v_inf_fb_3(:,i));
            % hyperbolic velocity at rp_targ
            v_per_hyp_arr(i,1) = sqrt(v_inf_arr(i,1)^2 + 2*mu_P2/rp_targ);
            % DV required to get into the target orbit
            DV_arr(i,1) = norm(vp_targ - v_per_hyp_arr(i,1));
```

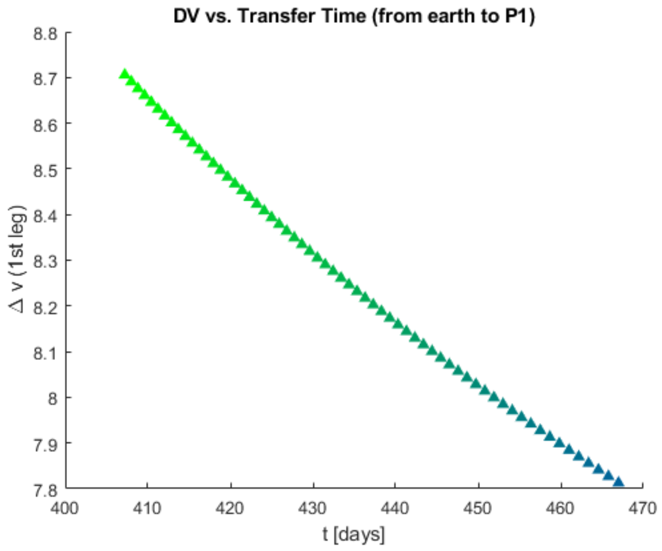
```
        else
            traj_color = [0.8,0.8,0.8];
            figure(1),hold on
            plot(X_sc1(:,1),X_sc1(:,2)    ,'linewidth',0.5,'linestyle','-','color',traj_color)
            plot(X_sc2(:,1),X_sc2(:,2)    ,'linewidth',0.5,'linestyle','-','color',traj_color)
            daspect([1,1,1])
            flag_stop = 1;
            break
        end
        i=i+1;
```

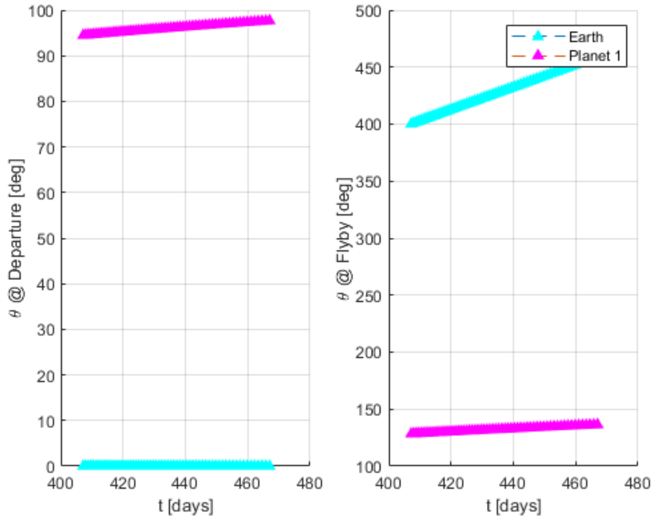
```
    else
        flag_stop = 1;
    end
```

```
end

N=i-1;
for i=1:N
    traj_color = [0,(i_max+1-i)/i_max,(i-1)/i_max];
    figure(2),hold on
    subplot(2,2,1),hold on
    plot(Total_Time(i,1)/3600/24, DV_dep(i,1), 'markeredgecolor',traj_color,'markerfacecolor',traj_color,'markersize',5.0,'marker','^','linestyle','--')

    subplot(2,2,3),hold on
    plot(Total_Time(i,1)/3600/24, DV_arr(i,1), 'markeredgecolor',traj_color,'markerfacecolor',traj_color,'markersize',5.0,'marker','>','linestyle','--')

    subplot(2,2,[2,4]),hold on
    plot(Total_Time(i,1)/3600/24, DV_dep(i,1)+DV_arr(i,1), 'markeredgecolor',traj_color,'markerfacecolor',traj_color,'markersize',4.0,'marker','s','linestyle','--')
end

% Plots of DVs vs total transfer time
Time_min = min(Total_Time(:,1))/3600/24;
if max(Total_Time(:,1))>T_Hohmann_EP2
    Time_max = max(Total_Time(:,1))/3600/24;
else
    Time_max = T_Hohmann_EP2/3600/24;
end

Time_dep_extr = spline(DV_dep,Total_Time/3600/24);
yy_dep = ppval(Time_dep_extr, linspace(min(DV_dep),max(DV_dep),5000));

Time_arr_extr = spline(DV_arr,Total_Time/3600/24);
yy_arr = ppval(Time_arr_extr, linspace(min(DV_arr),max(DV_arr),5000));

DV_tot = DV_dep+DV_arr;
Time_tot_extr = spline(DV_tot,Total_Time/3600/24);
yy_tot = ppval(Time_tot_extr, linspace(min(DV_tot),max(DV_tot),5000));


figure(2),hold on
subplot(2,2,1),hold on,grid on
xlabel('Transfer Time [days]')
ylabel('\Delta V (departure)  [km/s]')
plot(T_Hohmann_EP2/3600/24, DV_dep_Hohmann, 'markeredgecolor','r','markerfacecolor','r','markersize',5.0,'marker','^')
plot(yy_dep,linspace(min(DV_dep),max(DV_dep),5000),'--k')
line([Time_min, Time_max],[DV_dep_Hohmann, DV_dep_Hohmann],'linestyle','--','color','r')

subplot(2,2,3),hold on,grid on
xlabel('Transfer Time [days]')
ylabel('\Delta V (arrival)  [km/s]')
plot(T_Hohmann_EP2/3600/24, DV_arr_Hohmann, 'markeredgecolor','r','markerfacecolor','r','markersize',5.0,'marker','>')
plot(yy_arr,linspace(min(DV_arr),max(DV_arr),5000),'--k')
line([Time_min, Time_max],[DV_arr_Hohmann, DV_arr_Hohmann],'linestyle','--','color','r')

subplot(2,2,[2,4]),hold on,grid on
xlabel('Transfer Time [days]')
ylabel('\Delta V (total)  [km/s]')
plot(T_Hohmann_EP2/3600/24, DV_tot_Hohmann, 'markeredgecolor','r','markerfacecolor','r','markersize',5.0,'marker','s')
plot(yy_tot,linspace(min(DV_tot),max(DV_tot),5000),'--k')
line([Time_min, Time_max],[DV_tot_Hohmann, DV_tot_Hohmann],'linestyle','--','color','r')
txt = 'Hohmann Transfer';
text(T_Hohmann_EP2/3600/24, DV_tot_Hohmann+0.1,txt,'HorizontalAlignment','Right')


figure(1),hold on
title('Interplanetary Trajectories')
xlabel('x [km]')
ylabel('y [km]')
plot(X_sc_Hohmann(:,1),X_sc_Hohmann(:,2)    ,'linewidth',2.0,'linestyle','--','color','r')
txt_Sun     = 'Sun';    text(-3E7,    0    ,txt_Sun)
txt_Earth   = 'Earth';  text(-3E7, -r_Earth,txt_Earth)
txt_P1 = Planet_1;      text(-3E7, -r_P1   ,txt_P1)
txt_P2 = Planet_2;      text(-3E7, -r_P2   ,txt_P2)

figure(3),hold on
title('DV vs. Transfer Time (from earth to P1)')
for i=1:N
    traj_color = [0,(i_max+1-i)/i_max,(i-1)/i_max];
    xlabel('t [days]')
    ylabel('\Delta v (1st leg)')
    plot(Transfer_Time1(i,1)/3600/24, DV_dep(i,1), 'markeredgecolor',traj_color,'markerfacecolor',traj_color,'markersize',5.0,'marker','^','linestyle','--')
end
for i=1:N
    traj_color = [0,(i_max+1-i)/i_max,(i-1)/i_max];
    figure(4),hold on,grid on

    subplot(1,2,1),hold on,grid on
    xlabel('t [days]')
    ylabel('\theta @ Departure [deg]')
    plot(Transfer_Time1(i,1)/3600/24, 0             , 'markeredgecolor','c','markerfacecolor','c','markersize',5.0,'marker','^','linestyle','--')
    plot(Transfer_Time1(i,1)/3600/24, theta_P1_dep(i,1), 'markeredgecolor','m','markerfacecolor','m','markersize',5.0,'marker','^','linestyle','--')


    subplot(1,2,2),hold on,grid on
    xlabel('t [days]')
    ylabel('\theta @ Flyby [deg]')
    plot(Transfer_Time1(i,1)/3600/24, theta_E_arr(i,1),  'markeredgecolor','c','markerfacecolor','c','markersize',5.0,'marker','^','linestyle','--')
    plot(Transfer_Time1(i,1)/3600/24, theta_P1_arr(i,1), 'markeredgecolor','m','markerfacecolor','m','markersize',5.0,'marker','^','linestyle','--')

    legend('Earth','Planet 1')
end
```









Predicting mission Total Time and DV

```
%time = input('Insert desired transfer time in days: ');
time = 3558.75;
DV_dep_predicted = spline(Total_Time/3600/24, DV_dep, time);
DV_arr_predicted = spline(Total_Time/3600/24, DV_arr, time);
DV_tot_predicted = spline(Total_Time/3600/24, DV_tot, time);

fprintf('\nPredicted DV for an interplanetary transfer of %d days:\n',time)
fprintf('                 DV(dep.) = %.3f km/s\n',DV_dep_predicted)
fprintf('                 DV(arr.) = %.3f km/s\n',DV_arr_predicted)
fprintf('                 DV(tot.) = %.3f km/s\n\n\n',DV_tot_predicted)

%DV = input('Insert desired DV in km/s: ');
DV = DV_tot_predicted;
Total_Time_predicted = spline(DV_tot,Total_Time/3600/24, DV);

fprintf('\nPredicted time for an interplanetary transfer with DV = %.3f km/s:\n',DV)
fprintf('               Total Time = %.0f  days\n',Total_Time_predicted)
```

```
Predicted DV for an interplanetary transfer of 3.558750e+03 days:
                 DV(dep.) = 8.010 km/s
                 DV(arr.) = 3.476 km/s
                 DV(tot.) = 11.486 km/s



Predicted time for an interplanetary transfer with DV = 11.486 km/s:
               Total Time = 3559  days
```

Published with MATLAB® R2020b