Homework 5 -- B-Plane targeting

```
%{
Problem 1 -- Make plot of requested partials as functions of the
perturbation size, allowing perturbations to range from 10 km/s to
 5e-16
km/s, use log scale on x-axis
%}
clear all;close all;clc
mu_E = 3.986004415e5;
r_SOI = [546507.344255845;-527978.380486028;531109.066836708];
v_SOI = [-4.9220589268733;5.36316523097915;-5.22166308425181];
%Calc. B-Plane parameters
k = [0 0 1];
h_hat = (cross(r_SOI,v_SOI))/(norm(cross(r_SOI,v_SOI)));
ehat = (1/mu_E)*((norm(v_SOI)^2 - mu_E/norm(r_SOI))*r_SOI -
 (dot(r_SOI,v_SOI))*v_SOI);
e = norm(ehat);
rho = acos(1/e);
S_hat = cos(rho)*(ehat/norm(ehat)) + sin(rho)*(cross(h_hat,ehat))/
(norm(cross(h_hat,ehat)));
T_hat = (cross(S_hat,k))/norm(cross(S_hat,k));
R_hat = cross(S_hat,T_hat);
B_hat = cross(S_hat,h_hat);
[a,ecc,inc,w,Omega,P,ehat,ehatperp,f] = compOE(r_SOI,v_SOI,mu_E);

a = abs(a);
c = a*e;
b = a*sqrt(e^2 - 1);

B = b*B_hat;
B_T = dot(B,T_hat);
B_R = dot(B,R_hat);
theta = acos(dot(T_hat,B_hat));
if B_R < 0
    theta = 2*pi - theta;
end

B_Rnom = B_R;
B_Tnom = B_T;

pert = logspace(-16,1,500);
v_SOI_nom = [-4.9220589268733;5.36316523097915;-5.22166308425181];

for i = 1:numel(pert)
    %adjust x velocity component
    r_SOI = [546507.344255845;-527978.380486028;531109.066836708];
    v_SOI_pert = v_SOI_nom + [pert(i);0;0];
    %Calc. B-Plane parameters
    k = [0 0 1];
    h_hat = (cross(r_SOI,v_SOI_pert))/(norm(cross(r_SOI,v_SOI_pert)));
    ehat = (1/mu_E)*((norm(v_SOI_pert)^2 - mu_E/norm(r_SOI))*r_SOI -
 (dot(r_SOI,v_SOI_pert))*v_SOI_pert);
    e = norm(ehat);
    rho = acos(1/e);
    S_hat = cos(rho)*(ehat/norm(ehat)) + sin(rho)*(cross(h_hat,ehat))/
(norm(cross(h_hat,ehat)));
    T_hat = (cross(S_hat,k))/norm(cross(S_hat,k));
    R_hat = cross(S_hat,T_hat);
    B_hat = cross(S_hat,h_hat);
    [a,ecc,inc,w,Omega,P,ehat,ehatperp,f] =
 compOE(r_SOI,v_SOI_pert,mu_E);

    a = abs(a);
    c = a*e;
    b = a*sqrt(e^2 - 1);

    B = b*B_hat;
    B_T = dot(B,T_hat);
    B_R = dot(B,R_hat);

    J(1,1,i) = (B_T - B_Tnom)/pert(i);
    dBTdDx(i) = (B_T - B_Tnom)/pert(i);
    J(2,1,i) = (B_R - B_Rnom)/pert(i);
    dBRdDx(i) = (B_R - B_Rnom)/pert(i);
    %adjust y velocity component
    r_SOI = [546507.344255845;-527978.380486028;531109.066836708];
    v_SOI_pert = v_SOI_nom + [0;pert(i);0];
    %Calc. B-Plane parameters
    k = [0 0 1];
    h_hat = (cross(r_SOI,v_SOI_pert))/(norm(cross(r_SOI,v_SOI_pert)));
    ehat = (1/mu_E)*((norm(v_SOI_pert)^2 - mu_E/norm(r_SOI))*r_SOI -
 (dot(r_SOI,v_SOI_pert))*v_SOI_pert);
    e = norm(ehat);
    rho = acos(1/e);
    S_hat = cos(rho)*(ehat/norm(ehat)) + sin(rho)*(cross(h_hat,ehat))/
(norm(cross(h_hat,ehat)));
    T_hat = (cross(S_hat,k))/norm(cross(S_hat,k));
    R_hat = cross(S_hat,T_hat);
    B_hat = cross(S_hat,h_hat);
    [a,ecc,inc,w,Omega,P,ehat,ehatperp,f] =
 compOE(r_SOI,v_SOI_pert,mu_E);

    a = abs(a);
    c = a*e;
    b = a*sqrt(e^2 - 1);

    B = b*B_hat;
    B_T = dot(B,T_hat);
    B_R = dot(B,R_hat);

    J(1,2,i) = (B_T - B_Tnom)/pert(i);
    dBTdDy(i) = (B_T - B_Tnom)/pert(i);
    J(2,2,i) = (B_R - B_Rnom)/pert(i);
    dBRdDy(i) = (B_R - B_Rnom)/pert(i);
```

```
end
figure
semilogx(pert,dBTdDx)
hold on
semilogx(pert,dBRdDx)
semilogx(pert,dBTdDy)
semilogx(pert,dBRdDy)
grid on
grid minor
xlim([1e-15 1e1])
xlabel('Perturbation Magnitude (km/s)')
ylabel('Numerical Partial Value')
title('Partial Values VS Perturbation')
legend('$\frac{\partial B_T}{\partial \Delta V_x}$','$
\frac{\partial B_R}{\partial \Delta V_x}$','$\frac{\partial B_T}
{\partial \Delta V_y}$','$\frac{\partial B_R}{\partial \Delta
 V_y}$','interpreter','latex')
```

Problem 2

```
%{
Differentially corect the velocity to achieve the desired B_plane
parameters. z-component of velocity remains fixed, and TOF can vary
%}
clear all;close all;clc
% pert = logspace(-12,-1,300);
pert = 1e-10;
mu_E = 3.986004415e5;
r_SOI = [546507.344255845;-527978.380486028;531109.066836708];
v_SOI_nom(:,1) =
 [-4.9220589268733;5.36316523097915;-5.22166308425181];

B_RD = 5022.26511510685;
B_TD = 13135.7982982557;

for i = 1:1000
    %Calc. B-Plane parameters
    k = [0 0 1];
    h_hat = (cross(r_SOI,v_SOI_nom(:,i)))/
(norm(cross(r_SOI,v_SOI_nom(:,i))));
    ehat = (1/mu_E)*((norm(v_SOI_nom(:,i))^2 - mu_E/norm(r_SOI))*r_SOI
 - (dot(r_SOI,v_SOI_nom(:,i)))*v_SOI_nom(:,i));
    e = norm(ehat);
    rho = acos(1/e);
    S_hat = cos(rho)*(ehat/norm(ehat)) + sin(rho)*(cross(h_hat,ehat))/
(norm(cross(h_hat,ehat)));
    T_hat = (cross(S_hat,k))/norm(cross(S_hat,k));
    R_hat = cross(S_hat,T_hat);
    B_hat = cross(S_hat,h_hat);
    [a,ecc,inc,w,Omega,P,ehat,ehatperp,f] =
 compOE(r_SOI,v_SOI_nom(:,i),mu_E);

    a = abs(a);
    c = a*e;
    b = a*sqrt(e^2 - 1);

    B = b*B_hat;
    B_T = dot(B,T_hat);
    B_R = dot(B,R_hat);
    theta = acos(dot(T_hat,B_hat));
    if B_R < 0
        theta = 2*pi - theta;
    end

    B_Rnom = B_R;
    B_Tnom = B_T;

%     pert = logspace(-12,-1,300);
%     v_SOI_nom(:,1) =
 [-4.9220589268733;5.36316523097915;-5.22166308425181];


    %adjust x velocity component
    r_SOI = [546507.344255845;-527978.380486028;531109.066836708];
    v_SOI_pert = v_SOI_nom(:,i) + [pert;0;0];
    %Calc. B-Plane parameters
    k = [0 0 1];
    h_hat = (cross(r_SOI,v_SOI_pert))/(norm(cross(r_SOI,v_SOI_pert)));
    ehat = (1/mu_E)*((norm(v_SOI_pert)^2 - mu_E/norm(r_SOI))*r_SOI -
 (dot(r_SOI,v_SOI_pert))*v_SOI_pert);
    e = norm(ehat);
    rho = acos(1/e);
    S_hat = cos(rho)*(ehat/norm(ehat)) + sin(rho)*(cross(h_hat,ehat))/
(norm(cross(h_hat,ehat)));
    T_hat = (cross(S_hat,k))/norm(cross(S_hat,k));
    R_hat = cross(S_hat,T_hat);
    B_hat = cross(S_hat,h_hat);
    [a,ecc,inc,w,Omega,P,ehat,ehatperp,f] =
 compOE(r_SOI,v_SOI_pert,mu_E);

    a = abs(a);
    c = a*e;
    b = a*sqrt(e^2 - 1);

    B = b*B_hat;
    B_T = dot(B,T_hat);
    B_R = dot(B,R_hat);

    J(1,1,i) = (B_T - B_Tnom)/pert;
    dBTdDx(i) = (B_T - B_Tnom)/pert;
    J(2,1,i) = (B_R - B_Rnom)/pert;
    dBRdDx(i) = (B_R - B_Rnom)/pert;
    %adjust y velocity component
    r_SOI = [546507.344255845;-527978.380486028;531109.066836708];
    v_SOI_pert = v_SOI_nom(:,i) + [0;pert;0];
    %Calc. B-Plane parameters
    k = [0 0 1];
    h_hat = (cross(r_SOI,v_SOI_pert))/(norm(cross(r_SOI,v_SOI_pert)));
    ehat = (1/mu_E)*((norm(v_SOI_pert)^2 - mu_E/norm(r_SOI))*r_SOI -
 (dot(r_SOI,v_SOI_pert))*v_SOI_pert);
    e = norm(ehat);
    rho = acos(1/e);
    S_hat = cos(rho)*(ehat/norm(ehat)) + sin(rho)*(cross(h_hat,ehat))/
(norm(cross(h_hat,ehat)));
    T_hat = (cross(S_hat,k))/norm(cross(S_hat,k));
    R_hat = cross(S_hat,T_hat);
    B_hat = cross(S_hat,h_hat);
    [a,ecc,inc,w,Omega,P,ehat,ehatperp,f] =
 compOE(r_SOI,v_SOI_pert,mu_E);

    a = abs(a);
    c = a*e;
    b = a*sqrt(e^2 - 1);

    B = b*B_hat;
    B_T = dot(B,T_hat);
    B_R = dot(B,R_hat);

    J(1,2,i) = (B_T - B_Tnom)/pert;
    dBTdDy(i) = (B_T - B_Tnom)/pert;
    J(2,2,i) = (B_R - B_Rnom)/pert;
    dBRdDy(i) = (B_R - B_Rnom)/pert;

    db = [B_TD - B_Tnom;B_RD - B_Rnom];
    dV(:,i) = inv(J(:,:,i))*db;

    v_SOI_nom(:,i+1) = v_SOI_nom(:,i) + [dV(:,i);0];

    if (norm(dV(:,i)) < 1e-6) && (abs(db(1)) < 1e-6) && (abs(db(2)) <
 1e-6)
        break
    end
end
dV_TCM = sum(dV,2);
```

Published with MATLAB® R2020b