## Table of Contents

```
clear all;close all;clc
```

# ASEN 6008 HW 4 -- Question 1

```
V_sc_sun = [-10.8559 -35.9372]; %km/s
V_venus_sun = [-15.1945 -31.7927]; %km/s
R_venus_sun = [-96948447.3751 46106976.1901]; %km

mu_sun = 1.32712440018e11; %km^3/s^2
mu_venus = 324858.599; %km^3/s^2

R_venus = 6051.8; %km

% part a -- compute energy w.r.t. sun before flyby
V_inf_in = (V_sc_sun - V_venus_sun);
mech_en = ((norm(V_sc_sun)^2)/2) - mu_sun/norm(R_venus_sun);
% part b -- plot turn angle for different values of r_p, vary r_p from
 0 to
% 200,000

r_p = [0:200000];

psi = pi - (2.*acos(1./(1 + norm(V_inf_in)^2.*(r_p./mu_venus))));

figure
plot(r_p,rad2deg(psi))
xlabel('r_p (km)')
ylabel('\Psi (rads)')
grid on
grid minor
title('Turn Angle \Psi VS r_p')
% parb c -- plot energy w.r.t to sun after flyby for different values
 of
% closest approach. Include plots for leading and trailing passes of
 planet.
% Include line on plot indicating where r_p = r_venus and line
 indicating
% the energy of the s/c w.r.t the sun before flyby
% C = [cos(psi) -sin(psi);sin(psi) cos(psi)]; % DCM to rotate incoming
 v_inf by psi to get outgoing v_inf
for i = 1:numel(psi)
    C = [cos(psi(i)) -sin(psi(i));sin(psi(i)) cos(psi(i))];
    V_inf_out = C'*V_inf_in';
    V = V_inf_out + V_venus_sun';
    en_post_flyby(i) = (norm(V)^2/2) - mu_sun./(norm(R_venus_sun) +
 r_p(i));
end
r_ven_line_t = R_venus*ones(1,1000);
r_ven_line = linspace(-750,-350,1000);
pre_flyby_en = mech_en*ones(1,numel(r_p));


figure
hold on
plot(r_p,en_post_flyby)
plot(r_ven_line_t,r_ven_line,'--k')
plot(r_p,pre_flyby_en,'--')
xlabel('r_p (km)')
ylabel('Energy \epsilon')
grid on
grid minor
title('Leading Pass Energy Post-Flyby VS r_p')
legend('Post-Flyby Energy','r_p = r_{Venus}','Pre-Flyby Energy')

for i = 1:numel(psi)
    C = [cos(psi(i)) -sin(psi(i));sin(psi(i)) cos(psi(i))];
    V_inf_out = C*V_inf_in';
    V = V_inf_out + V_venus_sun';
    en_post_flyby(i) = (norm(V)^2/2) - mu_sun./(norm(R_venus_sun) +
 r_p(i));
end
r_ven_line = linspace(-900,-300,1000);

figure
hold on
plot(r_p,en_post_flyby)
plot(r_ven_line_t,r_ven_line,'--k')
plot(r_p,pre_flyby_en,'--')
xlabel('r_p (km)')
ylabel('Energy \epsilon')
grid on
grid minor
title('Trailing Pass Energy Post-Flyby VS r_p')
legend('Post-Flyby Energy','r_p = r_{Venus}','Pre-Flyby Energy')
% plots seem to tell which closest approach (r_p) value gives you the
% lowest energy flybys and in turn helps to plan which r_p you should
 aim
% for given the desired outcome of a flyby
```

# Question 2 -- calculate r_p, Psi, B_T, B_R, norm(B), and theta

```
clear all;close all;clc

V_inf_in = [-5.19425 5.19424 -5.19425]; %km/s
```

```
V_inf_out = [-8.58481 1.17067 -2.42304]; %km/s
V_inf = norm(V_inf_in);
mu_E = 3.986004415e5; %km^3/s^2

k_hat = [0 0 1];
S_hat = V_inf_in/norm(V_inf_in);
T_hat = cross(S_hat,k_hat)/norm(cross(S_hat,k_hat));
R_hat = cross(S_hat,T_hat);
h_hat = cross(V_inf_in,V_inf_out)/norm(cross(V_inf_in,V_inf_out));
B_hat = cross(S_hat,h_hat);

a = -mu_E/V_inf^2;
num = dot(V_inf_out,V_inf_in);
denom = norm(V_inf_out)*norm(V_inf_in);
Psi = acos(num/denom);
rho = (pi - Psi)/2;
rp = (mu_E/V_inf^2)*((1/cos(rho))-1);
c = a/cos(rho);
b = sqrt(a^2*((c/a)^2 - 1));
B = b*B_hat;

theta = acos((dot(B_hat,T_hat))/(norm(B_hat)*norm(T_hat)));
B_T = dot(B_hat,T_hat);
B_R = dot(B_hat,R_hat);
normB = norm(B);
```

# Question 3

```
clear all;close all;clc
% compute altitude of closest approach for Venus flyby using Lambert's
% problem and 3D gravity flyby eqns.
mu_E = 3.986004415e5; %km^3/s^2
mu_V = 3.24858599e5;
mu_S = 1.32712440018e11;
R_venus = 6051.8; %Venus radii in km

Launch_JD = 2447807.5;
Venus_Flyby_JD = 2447932.5;
Earth_Flyby_JD = 2448235.5;

Launch_Ephem = Meeus(Launch_JD);
Flyby_Ephem = Meeus(Venus_Flyby_JD);

[r0E,v0E] =
 calcposvel(Launch_Ephem.Earth.a,Launch_Ephem.Earth.e,Launch_Ephem.Earth.i,Launch_
[r0V,v0V] =
 calcposvel(Flyby_Ephem.Venus.a,Flyby_Ephem.Venus.e,Flyby_Ephem.Venus.i,Flyby_Ephe

[v0,vf] = solvelambert(r0E,r0V,(Venus_Flyby_JD-Launch_JD)*86400,0);

V_inf_in = vf - v0V;

E_Flyby_Ephem = Meeus(Earth_Flyby_JD);
```

```
[r0E_FB,v0E_FB] =
 calcposvel(E_Flyby_Ephem.Earth.a,E_Flyby_Ephem.Earth.e,E_Flyby_Ephem.Earth.i,E_Fl

[v0_V_FB,vf_E_FB] = solvelambert(r0V,r0E_FB,(Earth_Flyby_JD-
Venus_Flyby_JD)*86400,0);
V_inf_out = v0_V_FB - v0V;

num = dot(V_inf_in,V_inf_out);
denom = norm(V_inf_in)*norm(V_inf_out);
Psi = acos(num/denom);

rp = (mu_V/norm(V_inf_in)^2)*((1/(cos((pi-Psi)/2))) - 1);

flyby_alt = rp - R_venus;

preflyby_en = (norm(v0)^2)/2 - (mu_S/norm(r0E));

postflyby_en = (norm(v0_V_FB)^2)/2 - (mu_S/norm(r0V));
```

*Published with MATLAB® R2020b*