

```

% Spring 2024 AER E 351 Homework 07 Problem 3.)9.5.)
% Matthew Mehrstens
clear; clc; close all;

%% Constants
EARTH = 1;
JUPITER = 2;

%% Equations
emos2kmpsec = @(v) v * 1.495978e8 / sqrt(1.495978e8^3 / 1.327e11); % [km/s]
kmpsec2emos = @(v) v * sqrt(1.495978e8^3 / 1.327e11) / 1.495978e8; % [EMOS]
au2km = @(d) d * 1.495978e8; % [km]
km2au = @(d) d / 1.495978e8; % [km]

% Velocity in a circular orbit with radius r.
v_c_fn = @(mu,r) sqrt(mu ./ r); % [velocity]

% mu is the mu of the sun, r_1 is the radius of the inner orbit, and r_2 is
% the radius of the outer orbit. Returns [v_periapse,v_apoapse].
v_Hohmann_fn = @(mu,r_1,r_2) [...
    sqrt(mu .* (2 ./ r_1 - 1 ./ ((r_1 + r_2) ./ 2))), ...
    sqrt(mu .* (2 ./ r_2 - 1 ./ ((r_1 + r_2) ./ 2)))]; % [velocity]

% Assumes massless, circular orbits. mu is relative to the sun, r_1 is the
% radius of the inner orbit, and r_2 is the radius of the outer orbit.
% Returns [Deltav_1,Deltav_2] where Deltav_1 is the Deltav required at the
% inner orbit and Deltav_2 is the Deltav required at the outer orbit.
Deltav_Hohmann_fn = @(mu,r_1,r_2) ...
    abs(v_Hohmann_fn(mu,r_1,r_2) - v_c_fn(mu,[r_1,r_2])); % [velocity]

% Change in velocity from a hyperbolic fly-by maneuver.
Deltav_FB_fn = @(v_inf,v_s,r_p,r_s) ...
    2 .* v_inf ./ (1 + (v_inf ./ v_s).^2 .* (r_p ./ r_s)); % [velocity]

delta_fn = @(Deltav_FB,v_inf) 2 * asin(Deltav_FB / (2 * v_inf)); % [rad]

% Lambert's Equations Stuff
c_fn = @(r_1,r_2,theta) sqrt(r_1^2 + r_2^2 ...
    - 2 * r_1 * r_2 * cos(theta)); % [distance]
s_fn = @(r_1,r_2,c) (r_1 + r_2 + c) / 2; % [distance]

% Had to make a correction to alpha.
lamberts_eqn = @(mu,t_F,s,c,a) -sqrt(mu) * t_F ...
    + a^(3 / 2) * ((2 * pi - 2 * asin(sqrt(s / (2 * a)))) ...
    - 2 * asin(sqrt((s - c) / (2 * a)))) ...
    - (sin(2 * pi - 2 * asin(sqrt(s / (2 * a)))) ...
    - sin(2 * asin(sqrt((s - c) / (2 * a))))));

years2ctu = @(t) t * 2 * pi; % [ctu]

%% Given
G = 6.67259e-20; % [km^3/(kg*s^2)]

m_Earth = 5.974e24; % [kg]
m_Jupiter = 317.938 * m_Earth; % [kg]

mu_Sun = 1; % [cdu^3/ctu^2]
mu_Earth = G * m_Earth; % [km^3/s^2]

```

```

mu_Jupiter = G * m_Jupiter; % [km^3/s^2]

Earth_radius = 6.37812e3; % [km]
Jupiter_radius = 11.209 * Earth_radius; % [km]

r_Earth = 1; % [au]
r_Jupiter = 5.2; % [au]

r_p_Jupiter = Jupiter_radius; % [km]

v_s_Jupiter = 1.415; % [EMOS]

% Mean orbital speed of Jupiter relative to the sun
v_c_Jupiter = sqrt(mu_Sun / r_Jupiter); % [EMOS]

Earth_siderial = 365.256; % [days]
T_Saturn = 29 + 167 / Earth_siderial; % [years]
T_Saturn = years2ctu(T_Saturn); % [ctu]

%% (a)
Deltav_Hohmann = Deltav_Hohmann_fn(mu_Sun,r_Earth,r_Jupiter); % [EMOS]
v_inf_Jupiter = Deltav_Hohmann(JUPITER); % [EMOS]

Deltav_FB = Deltav_FB_fn(v_inf_Jupiter,v_s_Jupiter,km2au(r_p_Jupiter), ...
    km2au(Jupiter_radius)); % [EMOS]

delta = delta_fn(Deltav_FB,v_inf_Jupiter); % [rad]

alpha = asin(v_inf_Jupiter * sin(delta) / Deltav_FB); % [rad]

v_sv0 = sqrt(v_c_Jupiter^2 + v_inf_Jupiter^2 ...
    - 2 * v_c_Jupiter * v_inf_Jupiter * cos(pi - alpha)); % [EMOS]

v_e = sqrt(2) * v_c_Jupiter; % [EMOS]

%% (b)
% a of the hyperbolic solar system escape trajectory
a_Sun = (2 / r_Jupiter - v_sv0^2 / mu_Sun)^(-1); % [au]

cosgamma = (v_c_Jupiter^2 + v_sv0^2 - v_inf_Jupiter^2) ...
    / (2 * v_c_Jupiter * v_sv0); % []

h_Sun = r_Jupiter * v_sv0 * cosgamma; % [au*EMOS]

e_Sun = sqrt(1 - h_Sun^2 / (mu_Sun * a_Sun)); % []

f_Jupiter = acos(a_Sun / (r_Jupiter * e_Sun) * (1 - e_Sun^2) ...
    - 1 / e_Sun); % [rad]

%% (c)
r_1 = r_Jupiter; % [au]
r_2 = 9.5388; % [au]
f_Saturn = acos(a_Sun / (r_2 * e_Sun) * (1 - e_Sun^2) ...
    - 1 / e_Sun); % [rad]
theta = f_Saturn - f_Jupiter; % [rad]

c = c_fn(r_1,r_2,theta);
s = s_fn(r_1,r_2,c);

```

```
t = fsolve(@(t) lamberts_eqn(mu_Sun,t,s,c,a_Sun),1); % [ctu]
```

```
n_Saturn = 2 * pi / T_Saturn; % [rad/ctu]
```

```
transfer_Saturn = t * n_Saturn; % [rad]
```

```
beta = theta - transfer_Saturn; % [rad]
```

```
%% Output
```

```
fprintf( ...
    "Problem 3.)9.5.)\n" + ...
    "a.)\n" + ...
    "Deltav_Hohmann_Earth = %g EMOS = %g km/s\n" + ...
    "Deltav_Hohmann_Jupiter = %g EMOS = %g km/s\n" + ...
    "v_inf_Jupiter = %g EMOS = %g km/s\n" + ...
    "Deltav_FB = %g EMOS = %g km/s\n" + ...
    "delta = %g rad = %g°\n" + ...
    "v_sv0 = %g EMOS = %g km/s\n" + ...
    "v_e = %g EMOS = %g km/s\n\n", ...
    Deltav_Hohmann(EARTH),emos2kmpsec(Deltav_Hohmann(EARTH)), ...
    Deltav_Hohmann(JUPITER),emos2kmpsec(Deltav_Hohmann(JUPITER)), ...
    v_inf_Jupiter,emos2kmpsec(v_inf_Jupiter),Deltav_FB, ...
    emos2kmpsec(Deltav_FB),delta,rad2deg(delta),v_sv0, ...
    emos2kmpsec(v_sv0),v_e,emos2kmpsec(v_e));
fprintf( ...
    "b.)\n" + ...
    "a_Sun = %g au = %g km\n" + ...
    "h_Sun = %g au*EMOS\n" + ...
    "e_Sun = %g\n" + ...
    "f @ Jupiter = %g rad = %g°\n\n", ...
    a_Sun, au2km(a_Sun),h_Sun,e_Sun,f_Jupiter,rad2deg(f_Jupiter));
fprintf( ...
    "c.)\n" + ...
    "r_1 = %g au\n" + ...
    "r_2 = %g au\n" + ...
    "f @ Saturn = %g rad = %g°\n" + ...
    "theta = %g rad = %g°\n" + ...
    "c = %g au = %g km\n" + ...
    "s = %g au = %g km\n" + ...
    "t = %g ctu\n" + ...
    "n_Saturn = %g rad/ctu\n" + ...
    "transfer_Saturn = %g rad = %g°\n" + ...
    "beta = %g rad = %g°\n", ...
    r_1,r_2,f_Saturn,rad2deg(f_Saturn),theta,rad2deg(theta),c,au2km(c), ...
    s,au2km(s),t,n_Saturn,transfer_Saturn,rad2deg(transfer_Saturn), ...
    beta,rad2deg(beta));
```