Table of Contents

```
clear all
close all
% clc
addpath("C:\joshFunctionsMatlab\")

mu_e = 398600;
```

formating and t0 vecs

```
sv=load("stateVecs.mat");
a = sv.obj1;
b = sv.obj2;
c = sv.obj3;
d = sv.obj4;
obj = [a,b,c,d]; % generate struct

clear sv a b c d

t0 = obj(1).time; % julian date of object1 will be time0 for our mission

n = length(obj);

for i = 1:n
    obj(i).t = obj(i).time; % change field name
    obj(i).t = obj(i).t-t0; % get days relative to t0
    obj(i).t = obj(i).t*24*3600;%convert from solar days to seconds
end

t0 = 0; % we change t0 to be the reference point

obj = rmfield(obj,'time'); % change field name part 2

obj = updateCOES(obj);
```

```
for i = 1:n
    r = obj(i).r;
    v = obj(i).v;
    a = obj(i).a;
    dt = -obj(i).t;
    [r0,v0] = prop1(r,v,a,dt);
    obj(i).r0 = r0;
    obj(i).v0 = v0;
    obj(i).r = r0;
    obj(i).v = v0;
    obj(i).theta = nan; % theta isn't right anymore
end
t = t0; % currently its t0
clear r v r0 v0 dt a
% obj = rmfield(obj,'r');
% obj = rmfield(obj,'v');
obj = rmfield(obj,'t');
% obj = updateCOES(obj);
```

Time 1 - 5 periods of object1

```
dt = obj(1).T*5;

t = [t,dt]; % its now 5*T past T0

for i = 1:n
    r0 = obj(i).r;
    v0 = obj(i).v;
    a = obj(i).a;
%     [r,v] = prop1(r0,v0,a,dt)
    [r,v] = prop3(r0,v0,dt);
    obj(i).r = r;
    obj(i).v = v;
    obj(i).theta = nan; % theta isn't right anymore
end

obj = [obj,obj(1)];
obj(5).name = "poop";
n = n+1;
obj=updateCOES(obj);

clear r0 v0 a r v
```

calculate transfer 1

assume that for these first two orbits that they are circular. We will take the speeds and radii to be constant.

```
v1 = norm(obj(1).v);
v2 = norm(obj(2).v);

r1 = norm(obj(1).r);
r2 = norm(obj(2).r);
```

```
rtransfer = [3.77996066;-1.67356548;5.83338712]*10^6;
angleTillNode = acos(dot(obj(1).r,rtransfer)/(r1*norm(rtransfer))); % rad
arcLength = (angleTillNode*r1);
dt = arcLength/v1; % time till node

clear angleTillNode arcLength

t = [t,dt]; % its now 5*T+time till node past T0
%sum(t) %time since t0 to raan inc and homann time
for i = 1:n
    r0 = obj(i).r;
    v0 = obj(i).v;
    a = obj(i).a;
    [r,v] = prop3(r0,v0,dt); % go to inc raan change
    obj(i).r = r;
    obj(i).v = v;
    obj(i).theta = nan; % theta isn't right anymore
end
```

lets inc raan change

```
inc1 = obj(1).inc;
inc2 = obj(2).inc;
draan = obj(1).raan-obj(2).raan;
alpha = acos(cos(inc1)*cos(inc2)+sin(inc1)*sin(inc2)*cos(draan));
% clear inc1 inc2 daop
dv = 2*v1*sin(alpha/2);
```

lets homann

```
[dv1,dv2,~,dt] = joshHomann(r1,v1,r2,v2,mu_e);
dv = [dv,dv1,dv2];

t = [t,dt];
beforeHomannIncRaan = obj;
% sum(t) %time after homann
for i = 1:n
    r0 = obj(i).r;
    v0 = obj(i).v;
    a = obj(i).a;
    [r,v] = prop3(r0,v0,dt); % go to inc raan change
    obj(i).r = r;
    obj(i).v = v;
    obj(i).theta = nan; % theta isn't right anymore
end
obj(end).v = nan(3,1); % not right right anymore
obj(end).r = nan(3,1);


rArrive = r2*(-rtransfer/norm(rtransfer));
clear rtransfer r0 r1 r2 v0 v1 v2 inc1 inc2 dv1 dv2 i draan alpha a r v dt
```

phaser

```
phaseAngle = dot(rArrive,obj(2).r)/(norm(rArrive)*norm(obj(2).r));
clear rArrive
rcirc = norm(obj(2).r);
T = obj(2).T;

m = 10; % number of orbits
dT = T*(phaseAngle/(2*pi));
dT = dT/m;

Tp = T-dT;
ap = (Tp*( sqrt(mu_e)/(2*pi) ) )^(2/3);
rp = 2*ap-rcirc;
ecct = (rcirc-rp)/(rcirc+rp);
ht = sqrt(mu_e*rcirc*(1+ecct));

vat = ht/rcirc;
dvx = norm(obj(2).v)-vat;
dv = [dv,-dvx,dvx];

dt = m*Tp;

clear ecct dvx dT ap ht vat rcirc phaseAngle owensTimeAfterObj2 i a Tp

before10PhaseOrbits = obj;
%%%%%%%%%% propegate m orbits at 2 for phasing
for i = 1:n
    r0 = obj(i).r;
    v0 = obj(i).v;
    a = obj(i).a;
    [r,v] = prop3(r0,v0,dt);
    obj(i).r = r;
    obj(i).v = v;
    obj(i).theta = nan; % theta isn't right anymore
end
obj(end) = obj(2);
t = [t,dt];
clear v r v0 r0 dt
%%%%%%%%%%
```

lamberts town

% we'll assume that me and owen made it to the same place % owensTimeAfterObj2 = 7.0175e4; % dt = owensTimeAfterObj2;

```
before5orbitsAt2 = obj;
%%%%%%%%%% propegate 5 orbits at 2
dt = 5*obj(2).T;
for i = 1:n
    r0 = obj(i).r;
    v0 = obj(i).v;
    a = obj(i).a;
    [r,v] = prop3(r0,v0,dt);
    obj(i).r = r;
    obj(i).v = v;
    obj(i).theta = nan; % theta isn't right anymore
end
obj(end) = obj(2);
t = [t,dt];
clear v r v0 r0 dt
%%%%%%%%%%

beforeLam23coast = obj;
```

lamberts 2-3

```
dt1 = 9000; % time to wait
dt2 = 130000; % time to transfer

%%%%%%%%%% propegate some coast time

for i = 1:n
    r0 = obj(i).r;
    v0 = obj(i).v;
    a = obj(i).a;
    [r,v] = prop3(r0,v0,dt1);
    obj(i).r = r;
    obj(i).v = v;
    obj(i).theta = nan; % theta isn't right anymore
end

obj(end) = obj(2);
clear v r v0 r0
vobj1 = obj(2).v;
robj1 = obj(2).r;

beforeLam23flight = obj;

%%%%%%%%%% propegate flight time
for i = 1:n
    r0 = obj(i).r;
    v0 = obj(i).v;
    a = obj(i).a;
    [r,v] = prop3(r0,v0,dt2);
    obj(i).r = r;
    obj(i).v = v;
    obj(i).theta = nan; % theta isn't right anymore
end
obj(end) = obj(2);
clear v r v0 r0
%%%%%%%%%%
t = [t,dt1,dt2];

vobj2 = obj(3).v;
robj2 = obj(3).r;
```

```
% ready to run lamberts from obj2 to obj3
theta1 = acos(dot(robj1,robj2)/(norm(robj1)*norm(robj2)));
theta2 = 2*pi - acos(dot(robj1,robj2)/(norm(robj1)*norm(robj2)));

% set up stumpffys
coefs = 15;
[Cc,Sc]=joshStumpffCoeffs(coefs);
C = @(z) sum(Cc.*joshStumpffZ(z,coefs));
S = @(z) sum(Sc.*joshStumpffZ(z,coefs));


[fz,y,A,z,flag,glag,gdotlag] =
 joshfLambert(norm(robj1),norm(robj2),dt2,theta2,mu_e,Cc,Sc);
% [v1rb, v2rb] =lambertsRB(r1,r2,dt,1);

v1 = (1/glag)*(robj2-flag*robj1);
v2 = (1/glag)*(gdotlag*robj2-robj1);

dv1 = norm(v1-vobj1);
dv2 = norm(v2-vobj2);

dv = [dv,dv1,dv2];
clear dt1 dt2
```

coast at obj3 for 5 periods

```
after5periodsAt3 = obj;

dt = 5*obj(3).T;
for i = 1:n
    r0 = obj(i).r;
    v0 = obj(i).v;
    a = obj(i).a;
    [r,v] = prop3(r0,v0,dt);
    obj(i).r = r;
    obj(i).v = v;
    obj(i).theta = nan; % theta isn't right anymore
end
t = [t,dt];
obj(end) = obj(2);
clear v r v0 r0
```

lamberts 3-4

```
dt1 = 45000; % time to wait
dt2 = 25000; % time to transfer

%%%%%%%%%% propegate some coast time

beforeLam34coast = obj;

for i = 1:n
```

```
    r0 = obj(i).r;
    v0 = obj(i).v;
    a = obj(i).a;
    [r,v] = prop3(r0,v0,dt1);
    obj(i).r = r;
    obj(i).v = v;
    obj(i).theta = nan; % theta isn't right anymore
end

obj(end) = obj(2);
clear v r v0 r0
vobj1 = obj(3).v;
robj1 = obj(3).r;

beforeLam34flight = obj;

%%%%%%%%%% propegate flight time
for i = 1:n
    r0 = obj(i).r;
    v0 = obj(i).v;
    a = obj(i).a;
    [r,v] = prop3(r0,v0,dt2);
    obj(i).r = r;
    obj(i).v = v;
    obj(i).theta = nan; % theta isn't right anymore
end
obj(end) = obj(3);
clear v r v0 r0
%%%%%%%%%%

afterLambert34flight = obj;

t = [t,dt1,dt2];


vobj2 = obj(4).v;
robj2 = obj(4).r;

% ready to run lamberts from obj2 to obj3
theta1 = acos(dot(robj1,robj2)/(norm(robj1)*norm(robj2)));
theta2 = 2*pi - acos(dot(robj1,robj2)/(norm(robj1)*norm(robj2)));

% set up stumpffys
% coefs = 15;
% [Cc,Sc]=joshStumpffCoeffs(coefs);
% C = @(z) sum(Cc.*joshStumpffZ(z,coefs));
% S = @(z) sum(Sc.*joshStumpffZ(z,coefs));

[fz,y,A,z,flag,glag,gdotlag] =
 joshfLambert(norm(robj1),norm(robj2),dt2,theta2,mu_e,Cc,Sc);
% [v1rb, v2rb] =lambertsRB(r1,r2,dt,1);

v1 = (1/glag)*(robj2-flag*robj1);
v2 = (1/glag)*(gdotlag*robj2-robj1);
```

```
dv1 = norm(v1-vobj1);
dv2 = norm(v2-vobj2);

dv = [dv,dv1,dv2];
```

dv total

t dv

```
dvf = sum(abs(dv));
tf = sum(t);
```

functions

```
function obj = updateCOES(obj)
for i = 1:length(obj)
    r = obj(i).r;
    v = obj(i).v;
    mu_e = 398600;
    [a,ecc,theta,inc,raan,aop,h,T,E] = joshCOE(r,v,mu_e);
    obj(i).a = a;
    obj(i).ecc = ecc;
    obj(i).theta = theta;
    obj(i).inc = inc;
    obj(i).raan = raan;
    obj(i).aop = aop;
    obj(i).h = h;
    obj(i).T = T;
    obj(i).E = E;
end
end

function [r,v] = prop1(r0,v0,a,dt)
mu_e = 398600;
coefs = 15;
[Cc,Sc]=joshStumpffCoeffs(coefs);
C = @(z) sum(Cc.*joshStumpffZ(z,coefs));
S = @(z) sum(Sc.*joshStumpffZ(z,coefs));
[fX,fpX]=joshfChi(r0,v0,mu_e,a,dt,Cc,Sc);
X0 = sqrt(mu_e)*dt*abs(1/a);
[X] = joshNewtons(fX,fpX,X0,1e-14);

f = 1-(X^2/norm(r0))*C(X^2/a);
g = dt-((1/sqrt(mu_e))*X^3*S(X^2/a));

r = f*r0+g*v0; % position
f_dot = (sqrt(mu_e)/(norm(r)*norm(r0)))*X*((X^2/a)*S(X^2/a)-1);
g_dot = 1-(X^2/norm(r))*C(X^2/a);
v = f_dot*r0 + g_dot*v0; % velocity
end

% function [r,v] = prop2(r0,v0,a,dt)
```

```
%
% mu_e = 398600;
% coefs = 15;
% [Cc,Sc]=joshStumpffCoeffs(coefs);
% C = @(z) sum(Cc.*joshStumpffZ(z,coefs));
% S = @(z) sum(Sc.*joshStumpffZ(z,coefs));
%
% fX=joshfChi(r0,v0,mu_e,a,dt,Cc,Sc);
% X0 = sqrt(mu_e)*dt*abs(1/a);
% % [X] = joshNewtons(fX,fpX,X0,1e-14);
% X = fzero(fX,X0);
%
% f = 1-(X^2/norm(r0))*C(X^2/a);
% g = dt-((1/sqrt(mu_e))*X^3*S(X^2/a));
%
% r = f*r0+g*v0; % position
% f_dot = (sqrt(mu_e)/(norm(r)*norm(r0)))*X*((X^2/a)*S(X^2/a)-1);
% g_dot = 1-(X^2/norm(r))*C(X^2/a);
% v = f_dot*r0 + g_dot*v0; % velocity
% end

function [r,v] = prop3(r0,v0,dt)
    X0 = [r0;v0];
    options = odeset('RelTol', 1e-8,'AbsTol',1e-13);
    [~,X] = ode45(@orbitODEFun,[0,dt],X0,options);
    X = X(end,:);
    r = X(1:3)';
    v = X(4:6)';
end

function Xdot = orbitODEFun(t,X)
    mu_e = 398600;
    r = X(1:3);
    v = X(4:6);
    vdot = (-mu_e/norm(r)^3)*r;
    rdot = v;
    Xdot = [rdot;vdot];
end
```

Published with MATLAB® R2022a