Table of Contents


```
function SecularStudy
```

Simulation Parameters:

```
n = 10; % Simulate for Ten Periods
tspan = [0 n*2*pi];
```

ad = 0:

```
r0 = [.1172;.6828;.4]; v0 = [-1.0454;-.1794;.6124]; as = .001; ad =
 [0;0;as]; mu = 1;
x0 = [r0;v0];

[t1,x1] =
 ode45(@(t,x)EOM(t,x,ad,mu),tspan,x0,odeset('AbsTol',1e-12,'RelTol',1e-9));
a1 = CartesiantoOrbitalElements(x1',mu);

da = @(a,e,i,w,n,E,as) 2*sind(i)/n^2*(sqrt(1-e^2)*cosd(w)*sind(E)-
sind(w)*(1-cosd(E)))*as;
de = @(a,e,i,w,n,E,as) sqrt(1-e^2)/
(n^2*a)*sind(i)*(3/2*cosd(w)*E*pi/180-2*e*cosd(w)*sind(E)+1/4*cosd(w)*sind(2*E)+1/
e^2)*(cosd(2*E)-1))*as;
a(1) = a1(1,1);
e(1) = a1(2,1);
tae(1) = 0;
k = 1;
n = sqrt(a1(1,1)^3/mu);
for k = 1:t1(end)/(2*pi)
%     a(k+1) = a(k) +
 da(a1(1,1),a1(2,1),a1(3,1),a1(5,1),n,a1(6,k),as)*(t1(k+1)-t1(k));
%     e(k+1) = e(k) +
 de(a1(1,1),a1(2,1),a1(3,1),a1(5,1),n,a1(6,k),as)*(t1(k+1)-t1(k));
    [~,j] = min(abs(t1-n*2*pi*k));
    a(k+1) = a(k) + da(a1(1,j),a1(2,j),a1(3,j),a1(5,j),n,360,as)*n;
    e(k+1) = e(k) + de(a1(1,j),a1(2,j),a1(3,j),a1(5,j),n,360,as)*n;
    tae(k+1) = tae(k) + n*(2*pi);
    k = k + 1;
end

figure;
```

```
subplot(8,1,[1 2]); axis equal;
plot3(x1(:,1),x1(:,2),x1(:,3));
subplot(8,1,3); hold on;
plot(t1,a1(1,:));
plot(tae,a);
plot([t1(1) t1(end)],a1(1,1)-sind(a1(3,1))*sind(a1(5,1))*(a1(2,1)+2)/
n^2*as*ones(1,2));
legend('Actual','Secular','Average');
ylabel('a');
subplot(8,1,4); hold on;
plot(t1,a1(2,:));
plot(tae,e);
plot([t1(1) t1(end)],(a1(2,1)-
sind(a1(3,1))*sind(a1(5,1))*(a1(2,1)^2-1)/
(4*n^2*a1(1,1))*as)*ones(1,2));
legend('Actual','Secular','Average?');
ylabel('e');
subplot(8,1,5);
plot(t1,a1(3,:));
ylabel('i[deg]');
subplot(8,1,6);
plot(t1,a1(4,:));
ylabel('\Omega[deg]');
subplot(8,1,7);
plot(t1,a1(5,:));
ylabel('\omega[deg]');
subplot(8,1,8);
plot(t1,a1(6,:));
ylabel('M[deg]');
xlabel('Time');
sgtitle('2c/d');

end
```

EOM:

```
function xdot = EOM(t,x,ad,mu)

xdot = [x(4:6); -mu/norm(x(1:3))^3*x(1:3)+ad];

end
```

Orbital Elements Converter:

```
function a = CartesiantoOrbitalElements(x,mu)

% Calculate Norms and Vectors of r and v:
    v = x(4:6,:);
    r = x(1:3,:);
    vmag = sqrt(sum(v.^2));
    rmag = sqrt(sum(r.^2));

    % Vectors:
```

```
    h = cross(r,v); % Angular Momentum
    hmag = sqrt(sum(h.^2));
    n = cross([0;0;1]*ones(1,length(h)),h); % Node Vector
    nmag = sqrt(sum(n.^2));
    e = 1/mu*((vmag.^2-mu./rmag).*r-dot(r,v).*v); % Eccentricity
    %e = 1/mu*cross(v,h)-r/rmag;
    emag = sqrt(sum(e.^2));

    % Angles:
    i = acosd(h(3,:)./(hmag));
    Om = acosd(n(1,:)./nmag);
    Om(n(2,:)<0) = 360 - Om(n(2,:)<0);
    if nmag == 0
        om = acosd(e(1,:)./emag);
        om(emag<10^-8) = NaN;
    else
        om = acosd(dot(n,e)./(nmag.*emag));
    end
    om(e(3,:)<0) = 360 - om(e(3,:)<0);

    % Other:
    p = hmag.^2/mu;
    a = p./(1-emag.^2);
    f = acos(dot(e,r)./(emag.*rmag));
    f(dot(r,v)<0) = 2*pi - f(dot(r,v)<0);
    f(emag<10^-8) = NaN;

    M = 180/pi*(asin(sqrt(1-emag.^2).*sin(f)./(1+emag.*cos(f)))-
emag.*sqrt(1-emag.^2).*sin(f)./(1+emag.*cos(f)));
    a = [a;emag;i;Om;om;M];

end
```

Published with MATLAB® R2020b