ModSim Exercise 11

Yoonyoung Cho

December 1 2015

1 Gravity

$$\vec{r} = \vec{p_1} - \vec{p_2}$$

$$\vec{F_g} = -\frac{G m_e m_b}{\vec{r}^2}\hat{\mathbf{r}}$$

With the above relation, the implementation is trivial:

```
function res = Gravity(m1,p1,m2,p2)%corresponds to gravity_force_func
r = (p1-p2); %direction : from p2 to p1
G = 6.67384e-11; %m^3 / kg*s^@ Gravity Constant
f_g = G*m1*m2/norm(r)^2; %magnitude
res = f_g*r/norm(r); %force vector
end
```

2 Orbital Simulation

```
function [t,y] = simulateOrbit()
%position of the Sun is taken as origin
p_s = [0 0 0]; %m
%778e9 = distance from the Sun to Jupiter
p_j = [0 778e9 0]; %m
y0 = packParams(p_s',[0 0 0]', p_j', [13.06e3 0 0]');
[t,y] = ode45(@orbitFlow,0:86400:86400*4332,y0);

[~,~,p_j,~] = unpackParams(y(end,:));
%Verification
disp(p_j); %position of Jupiter after 4332 days
theta = atan2(y(:,8),y(:,7)); %angle
figure;
plot(t,theta);
hold on;
line(xlim,[theta(end) theta(end)],'Color','r');
xlabel('time (sec)');
ylabel('angle (rad)');
title('Orbital Angle of Jupiter');

figure;
if(nargout == 0)
                %comet(y(:,7),y(:,8));
                myComet(t,y(:,1),y(:,2),y(:,7),y(:,8));
```

```
        end
end

function res = packParams(p_s,v_s,p_j,v_j)
%format into column vector
res = [p_s v_s p_j v_j];
res = reshape(res,12,1);
end

function [p_s,v_s,p_j,v_j] = unpackParams(y)
%reassign column vector to 4 vectors
y = reshape(y,3,4);
p_s = y(:,1);
v_s = y(:,2);
p_j = y(:,3);
v_j = y(:,4);
end

function res = orbitFlow(t, y)
[p_s,v_s,p_j,v_j] = unpackParams(y);
%p_s = Sun Position, m
%v_s = Sun Velocity, m/s
%p_j = Jupiter Position, m
%v_j = Jupiter Velocity, m/s
m_s = 2.0e30; %Sun Mass, kg
m_j = 1.89e27;%Jupiter Mass, kg
f_g = Gravity(m_s,p_s,m_j,p_j); %dir = J->S
dv_s = -f_g/m_s;%opposite pull
dv_j = f_g/m_j;
dp_s = v_s;
dp_j = v_j;
res = packParams(dp_s,dv_s,dp_j,dv_j);
end
```

According to the simulation, the position of Jupiter after 4332 days was found to be $\vec{p_j} =< 8.64e10, 7.70e11, 0 >$; here, the discrepancy in the x value, which indicates the position of Jupiter went slightly over the beginning point, can be attributed to the assumption that the orbit of Jupiter is perfectly circular, in which case the mean velocity of Jupiter may not have been the harmonic value for the circular orbit.

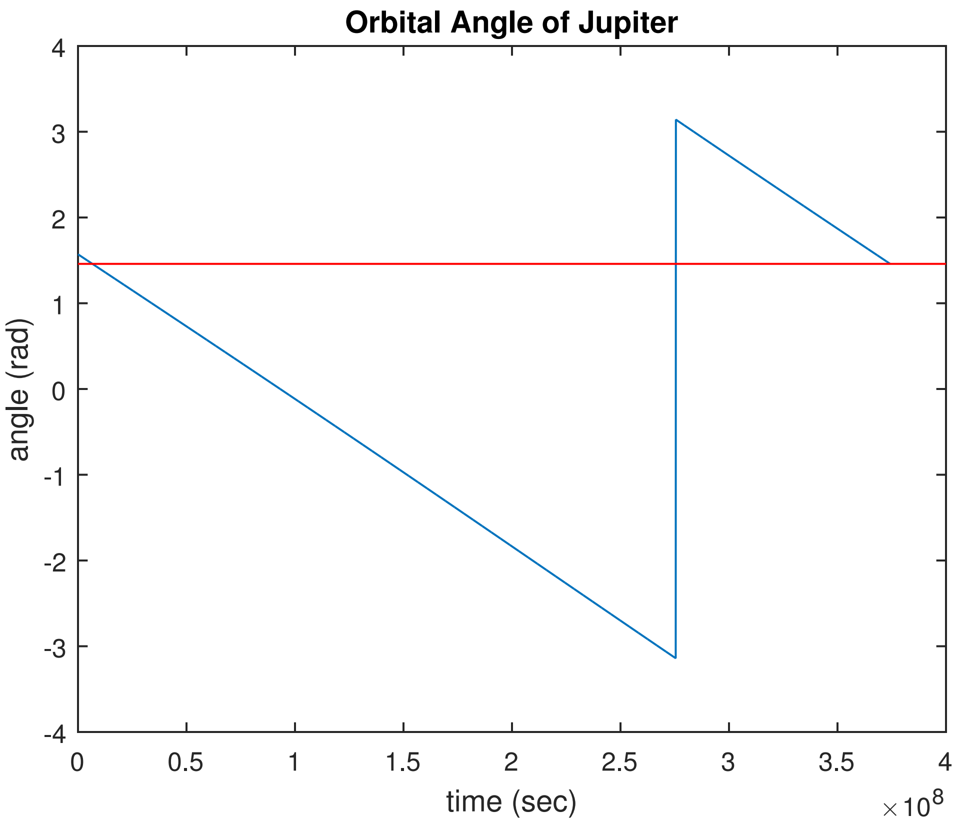

Figure 1: The Angle of Jupiter. The Sudden jump in the middle is the transition from $-\pi$ to π, which are equivalent.

Again, it is apparent that, in the course of 4332 days, Jupiter completes a cycle and exceeds the beginning angle by a bit.

3 Animation

```
function myComet(t,x_s,y_s,x_j,y_j)
% t = real orbit time(in sec)
        hold on;
        %scale duration to .001 sec/day
        duration = (max(t)-min(t)) / 86400 / 1000; %sec
        fps = 40; %frame/sec
        stepSize = floor(length(x_j)/(duration*fps));
        minmax = [min(x_j),max(x_j),min(y_j),max(y_j)]; %frame dimensions
        for i=1:stepSize:length(x_j) % # frames
                pause(1/fps);
                axis(minmax);
                plot(x_j(i),y_j(i),’r.’,’MarkerSize’,20);
                plot(x_s(i),y_s(i),’y.’,’MarkerSize’,50);
                drawnow;
        end
        xlabel(’x (m)’);
        ylabel(’y (m)’);
        title(’Orbit of Jupiter Around the Sun’);
```

```
end
```

Here, the pause duration was chosen such that the duration of the animation would be scaled to $.001 sec/day$:

$$
\begin{aligned}
dur &= time/1000/(60*60*24) \\
step &= \frac{length(x_j)}{dur*fps} \\
numFrame &= \frac{length(x_j)}{step}
\end{aligned}
$$

$$
\begin{aligned}
t &= \frac{1}{fps}*numFrame \\
&= \frac{1}{fps}*\frac{length(x_j)}{\frac{length(x_j)}{dur*fps}} \\
&= \frac{1}{fps}*dur*fps \\
&= dur
\end{aligned}
$$

A frame from the resultant animation is as follows:

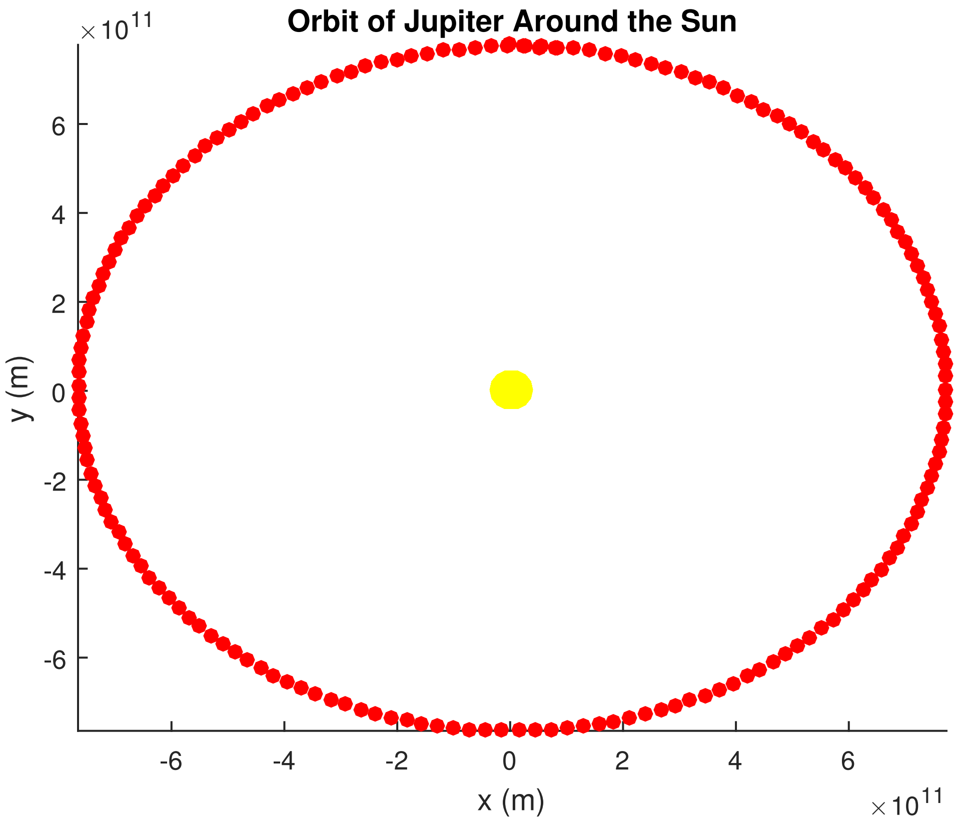

Figure 2: The orbit of Jupiter around the sun, after a cycle.