KATHMANDU UNIVERSITY

SCHOOL OF SCIENCE

DEPARTMENT OF APPLIED PHYSICS

Report on Simulation of Restricted Planar Three Body Problem

Submitted By:

Rabi Sharan Bogati (019237-15)

Anish Dhakal (019240-15)

Reema Shrestha (019243-15)

Supervised By:

Dr. Manish Pokharel

HOD of DOCSE

Submitted To:

Dr. Gajendra Sharma

Associate Professor of DOCSE

29th July, 2018

THIRD YEAR SECOND SEMESTER PROJECT REPORT

On

Simulation of Restricted Planar Three Body Problem

By:

Rabi Sharan Bogati (019237-15)
Anish Dhakal (019240-15)
Reema Shrestha (019243-15)

Approved by:

1. Project Supervisor

_____________________ ______________________________ ___________

(Signature) (Name) (Date)

2. Head/In-Charge of the Department

_____________________ ______________________________ ___________

(Signature) (Name) (Date)

Acknowledgment

We express our sincere thanks to our supervisor Mr. Manish Pokhrel who constantly supported and supervised our Project. We'd also like to thank Department of Applied Physics and Department of Computer Science and Engineering who provided us such a great opportunity. We'd also like to thank our teachers and seniors who helped us with the theoretical part of the project.

Abstract

The movement of a spacecraft between Earth and the Moon is an example of the infamous Three Body Problem. A general analytical solution for TBP is not known because of the complexity of solving the effect of three bodies which exerts on each other while moving, a total of six interactions. Mathematician Richard Arenstorf while at NASA solved a special case of this problem, by simplifying the interactions to four, because, the effect of the spacecraft's gravity upon the motion of the vastly more massive Earth and Moon is practically non-existent. Arenstorf found a stable orbit for a spacecraft orbiting between the Earth and Moon, shaped like an '8'.

In this project of ours, we are going to perform a simulation of this ‘8’ shaped orbit using Python.

Table of Contents

1. Introduction

1.1. Background

In physics and classical mechanics, the three-body problem is the problem of taking an initial set of data that specifies the positions, masses, and velocities of three bodies for some particular point in time and then determining the motions of the three bodies, in accordance with Newton's laws of motion and of universal gravitation, which are the laws of classical mechanics. The three-body problem is a special case of the *n*-body problem. Unlike two-body problems, there is no general closed-form solution for every condition, and numerical methods are needed to solve these problems.Historically, the first specific three-body problem to receive extended study was the one involving the Moon, the Earth, and the Sun. In an extended modern sense, a three-body problem is a class of problems in classical or quantum mechanics that models the motion of three particles.

1.2. Restricted Three Body Problem

The three-body problem has a special relevance, particularly in astrophysics and astrodynamics. In general, the three-body problem is classified into two types:

(i) The general three-body problem, which describes the motion of three bodies of arbitrary masses under their mutual attraction due to the gravitational field. The motion of the bodies takes place in three dimensions and there are no restrictions on their masses nor on the initial conditions.

(ii) The restricted three-body problem in which the mass of the third body is very small in comparison with the masses of the primaries, and it does not affect their motion. In this case, the primaries move around their center of mass either along circular or elliptical orbits.

Thus, the general problem has some applications in celestial mechanics such as the dynamics of triple star systems (and only a very few in space dynamics and solar system dynamics), whereas the restricted problem plays an important role in studying the motion of artificial satellites. It can be used also to evaluate the motion of the planets, minor planets and comets. The restricted problem gives an accurate description not only regarding the motion of the Moon but also with respect to the motion of other natural satellites. Furthermore, the restricted problem has many applications not only in celestial mechanics but also in physics, mathematics and quantum mechanics, to name a few. In quantum mechanics, a general form of the restricted problem is formed to solve the Schrödinger equation of helium-like ions. Furthermore, in modern solid state physics, the restricted problem can be used to discuss the motion of an infinitesimal mass affected not only by the gravitational field but also by light pressure from one (or both) of the primaries, which is called the photo gravitational problem.

The significance of the restricted problem when describing actual physical situations can be judged by the results obtained when these are compared with observations. It is worth mentioning that the utility might be prejudged by order of magnitude evaluations regarding the masses and the distances of the participating bodies.
In this way, a classic example in space dynamics is the Sun-Earth-Moon system.

2. Theory

2.1. The Arenstorf's '8' Shaped Orbit

The figure 8 design isn't because it is an optimal path. It occurs due to the gravity of moon and the Earth. When the spacecraft comes within the sphere of influence (SOI) of the moon, the spacecraft is pulled towards it. If the spacecraft is moving at escape velocity, the moon will perturb the flight but the spacecraft won't do a fly by. With the current speed, the moon's gravity is enough to cause an orbital fly by. Upon exiting the moon's SOI, the spacecraft is being pulled in by the Earth. Since the trajectory of the fly by was throwing the spacecraft away from the moon, it crosses its original path but this is short lived since the Earth then pulls it back in. If the spacecraft would have picked up enough velocity from the orbital maneuver to be on a parabolic or hyperbolic trajectory, it could have escaped the pull of Earth and been sent out into space.

One way to determine test speeds in designing a flight is to find the Jacobi constant, C. For a given C, the zero velocity curves are determined. Since we wanted to reach the moon, $C \geq -1.6649$ which corresponds to an initial velocity of at least 10.85762 but a velocity of 11.01 is the required escape velocity from the Earth so the initial speed has to be less than v_{esc}.

2.2. Equations of Motion

Derivations based on the book of Szebehely's Theory of Orbits.

$$r_{12} = \sqrt{(x_1 - x_2)^2}$$

$$x_2 = x_1 + r_{12}$$

$$x_1 = -\frac{m_2}{m_1 + m_2} r_{12}$$

x_1 is the $location\,of\ m_1\,relative\,to\,the\,center\,of\ gravity.$

$$\pi_1 = \frac{m_1}{m_1 + m_2}$$

$$x_2 = \frac{m_1}{m_1 + m_2} r_{12}$$

$$\pi_2 = \frac{m_2}{m_1 + m_2}$$

We can describe the position of m as $r = x\hat{i} + y\hat{j} + z\hat{k}$ in relation to the center of gravity, i.e., the origin.

$$r_1 = (x - x_1)\hat{i} + y\hat{j} + z\hat{k}$$

$$= (x + \pi_2 r_{12})\hat{i} + y\hat{j} + z\hat{k}$$

$$r_2 = (x - x_2)\hat{i} + y\hat{j} + z\hat{k}$$

$$= (x - \pi_1 r_{12})\hat{i} + y\hat{j} + z\hat{k}$$

Let's define the absolute acceleration where ω is the initial angular velocity which is constant. Then

$$\omega = \frac{2\pi}{T}.$$

$$\ddot{r}_{abs} = a_{rel} + a_{CG} + \Omega \times (\Omega \times r) + \dot{\Omega} \times r + 2\Omega \times v_{rel}$$

where,

$$a_{rel} = Rectilinear\ acceleration\ relative\ to\ the\ frame$$
$$\Omega \times (\Omega \times r) = Centripetal\ acceleration$$
$$2\Omega \times v_{rel} = Coriolis\ acceleration$$

Since the velocity of the center of gravity is constant, $a_{CG} = 0$, and $\dot{\Omega} = 0$ since the angular velocity of a circular orbit is constant. Therefore, the above equation becomes:

$$\ddot{r} = a_{rel} + \Omega \times (\Omega \times r) + 2\Omega \times v_{rel}$$

where,

$$\Omega = \Omega \hat{k}$$
$$r = x\hat{i} + y\hat{j} + z\hat{k}$$
$$\dot{r} = v_{CG} + \Omega \times r + v_{rel}$$
$$v_{rel} = \dot{x}\hat{i} + \dot{y}\hat{j} + \dot{z}\hat{k}$$
$$a_{rel} = \ddot{x}\hat{i} + \ddot{y}\hat{j} + \ddot{z}\hat{k}$$

Now we obtain

$$\ddot{r} = (\ddot{x} - 2\Omega\dot{y} - \Omega^2 x)\hat{i} + (\ddot{y} + 2\Omega\dot{x} - \Omega^2 y)\hat{j} + \ddot{z}\hat{k}.$$

Newton's second law of motion is $ma = F1 + F2$ where $F1 = -\frac{Gm_1 m}{r_1^3} r_1$ and $F2 = -\frac{Gm_2 m}{r_2^3} r_2$.

Let $\mu_1 = Gm_1$ and $\mu_2 = Gm_2$.

$m\mathbf{a} = \mathbf{F}1 + \mathbf{F}2$

$$ma = -\frac{m\mu_1}{r_1^3} r_1 - \frac{m\mu_2}{r_2^3} r_2$$

$$a = -\frac{\mu_1}{r_1^3} r_1 - \frac{\mu_2}{r_2^3} r_2$$

$$(\ddot{x} - 2\Omega\dot{y} - \Omega^2 x)\hat{i} + (\ddot{y} + 2\Omega\dot{x} - \Omega^2 y)\hat{j} + \ddot{z}\hat{k} = -\frac{\mu_1}{r_1^3} r_1 - \frac{\mu_2}{r_2^3} r_2$$

$$(\ddot{x} - 2\Omega\dot{y} - \Omega^2 x)\hat{i} + (\ddot{y} + 2\Omega\dot{x} - \Omega^2 y)\hat{j} + \ddot{z}\hat{k} = \begin{array}{l} -\frac{\mu_1}{r_1^3}[(x + \pi_2 r_{12})\hat{i} + \hat{j} + \hat{k}] \\ -\frac{\mu_2}{r_2^3}[(x - \pi_1 r_{12})\hat{i} + \hat{j} + \hat{k}] \end{array}$$

Now all we have to do is equate the coefficients.

$$(\ddot{x} - 2\Omega\dot{y} - \Omega^2 x) = -\frac{\mu_1}{r_1^3}(x + \pi_2 r_{12}) - \frac{\mu_2}{r_2^3}(x - \pi_1 r_{12})$$

$$\ddot{y} + 2\Omega\dot{x} - \Omega^2 y = -\frac{\mu_1}{r_1^3} y - \frac{\mu_2}{r_2^3} y$$

$$\ddot{z} = -\frac{\mu_1}{r_1^3} z - \frac{\mu_2}{r_2^3} z$$

We now have system of nonlinear ODEs. We can assume the trajectory is in plane and we do that by letting z=0 so we only have two equations remaining:

$$(\ddot{x} - 2\Omega\dot{y} - \Omega^2 x) = -\frac{\mu_1}{r_1^3}(x + \pi_2 r_{12}) - \frac{\mu_2}{r_2^3}(x - \pi_1 r_{12})$$

$$\ddot{y} + 2\Omega\dot{x} - \Omega^2 y = -\frac{\mu_1}{r_1^3} y - \frac{\mu_2}{r_2^3} y$$

Initial values are in the code below.

3. Coding Section

3.1 Python

Python is an interpreted high-level programming language for general-purpose programming. Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales.

3.2 Code

```
from vpython import*          #importing modules
from scipy.integrate import odeint
import numpy as np
import matplotlib.pyplot as plt

G=6.67259 * 10 ** -20      #values for variables
m1=5.974 * 10 ** 24
m2=7.348 * 10 ** 22
re=6378.0
rm=1737.0
r12 = 384400.0
M=m1+m2
pi1=m1/M
pi2=m2/M
u1=G*m1
u2=G*m2
u=u1+u2
w=np.sqrt((u/r12**3))
x1=-pi2*r12
x2=pi1*r12
d=200.0
phi=-np.pi*0.5
v0=10.9148
gamma=20*np.pi/180.0
r0=re+d
x=r0*np.cos(phi)+x1
y=r0*np.sin(phi)
vx=v0*(np.sin(gamma)*np.cos(phi)- np.cos(gamma)*np.sin(phi))
vy=v0* (np.sin(gamma)*np.sin(phi)+np.cos(gamma)*np.cos(phi))
t=0
tf=6.1*24*3600

def f(r,tpoints):      #define a function
    fvx=2*w*r[3]+w**2*r[0]-
(u1/(np.sqrt((r[0]+pi2*r12)**2+r[1]**2))**3)*(r[0]-x1)-(u2/(np.sqrt((r[0]-
pi1*r12)**2+r[1]**2))**3)*(r[0]-x2)
    fvy=-2*w*r[2]+(w**2-(u1/(np.sqrt((r[0]+pi2*r12)**2+r[1]**2))**3)-
(u2/(np.sqrt((r[0]-pi1*r12)**2+r[1]**2))**3))*r[1]
    return [r[2],r[3],fvx,fvy]

r=[x,y,vx,vy]              #solving ode
```

```
tpoints=np.linspace(t,tf,tf)
sol = odeint(f,r,tpoints)
plt.plot(sol[:,0],sol[:,1])
plt.show()

#make animation
scene=canvas(background=color.black,center=vector(24,3,0),width=700,height=35
0)
earth=sphere(radius=1,color=color.blue,pos=vector(0,0,0))
moon=sphere(radius=0.4,color=color.white,pos=vector(37.44,0,0))
sat=sphere(radius=0.2,color=color.orange,pos=vector(0,0,0),make_trail=True,tr
ail_type='points',trail_color=color.white)
label(pos=vector(24,14,0), text='Restricted planar three body
problem',box=False,color=color.red)
label(pos=vector(0,-2,0), text='earth',box=False)
label(pos=vector(37.44,-2,0), text='moon',box=False)
label(pos=vector(3,9,0), text='spacecraft trajectory',box=False)
time1=label(pos=vector(46,10,0),text="time")
position_x=label(pos=vector(46,8,0),text="X")
position_y=label(pos=vector(46,6,0),text="Y")
velocity=label(pos=vector(46,4,0),text="Velocity")
pointer = arrow(pos=vector(0,7,1),axis=vector(2,-2,-2), shaftwidth=0.3)
time=0
#making widgets
running = True
def Run(b):
    global running
    running =not running
    if running: b.text = "Run"
    else: b.text = "Pause"
button(text="Run", pos=vector(5,0,0), bind=Run)

while time<tf:
    rate(10000)
    if not running:
        p=vector((sol[time,0]/10000),(sol[time,1]/10000),0)
        p_x=sol[time,0]
        p_y=sol[time,1]
        v=(sol[time,2]**2+sol[time,3]**2)**(1/2)
        sat.pos=p
        time+=1

        time1.text=("Time:%.1f"%(time/60)+" min")
        position_x.text=("X:%.1f"%p_x)
        position_y.text=("Y:%.1f"%p_y)
        velocity.text=("velocity:%.1f"%v+"km/s")
```

4. Result and Conclusion

Finally using all these theories and codes, we finally were able to trace the orbit of spacecraft moving around the Earth and the Moon. The picture below clearly describes the restricted three body problem where the mass of spacecraft is taken as massless.

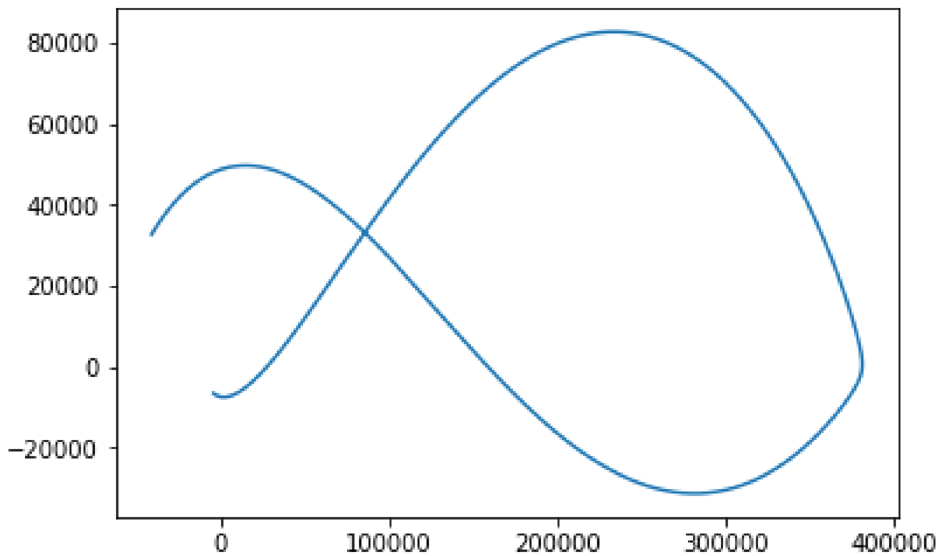

Fig: Plot of Restricted Planar Three Body Problem

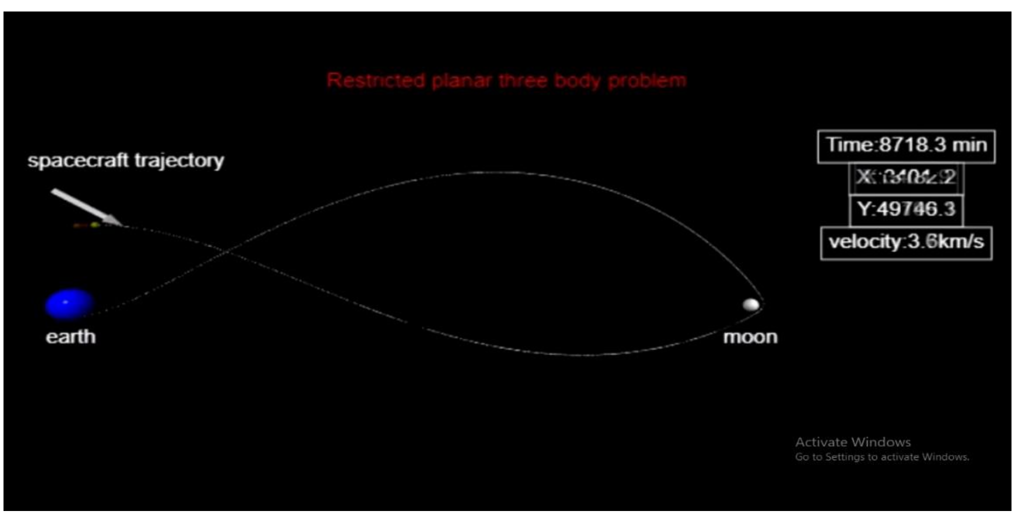

Fig: Simulation of Final Presentation

Gantt chart

Month	April	May	June	July
Literature Survey				
Proposal Submission				
Theory and Coding				
Report Submission and Presentation				

Bibliography

Broucke, R., Elipe, A., & Riaguas, A. (2006). *On the Figure-8 Periodic Solutions in the 3-Body Problem.*

EugeneI.Butikov. (n.d.). *MOTIONSOFCELESTIALBODIES: COMPUTERSIMULATIONS.*

Grégory Archambeau, P. A. (2011). *Eight-shaped Lissajous orbits in the Earth-Moon system.*

Guan, T. (n.d.). *Special cases of the three body problem.*

Szebehely, V. (1967). *Theory of Orbits, the Restricted Problem of Three Bodies.*

URLs

https://math.stackexchange.com/questions/54735/restricted-three-body-problem