FINDING SEMI-STABLE ORBITS AROUND A RING-SHAPED PLANETOID

Ada Matthews[1]

[1]Greenville University

I. Introduction

In an effort to find stable orbits around a torus-shaped planetoid, I explore the mathematics governing the restricted N-body system and simulate the dynamics numerically in Python.

This is the bare bones outline of a larger paper, I think, but this is what I have so far. The language used is also fairly conversational at times, which would need to be stripped away for a more formal research project.

II. Theory

In order to simplify calculations, we treat the system as a restricted N-Body problem. The Torus is discretized into N stationary point-masses positioned in a ring in the xy-plane centered at the origin.

The system of differential equations governing the behavior of this system, according to Newtonian mechanics, is

$$\ddot{\mathbf{r}} = GM_T \sum_{k=0}^{N} \frac{(\mathbf{r}_k - \mathbf{r})}{\|\mathbf{r}_k - \mathbf{r}\|^3}$$

a system of three, symmetrical, 2nd order differential equations, where $G = 6.67e-11(\frac{m^3}{kgs^2})$ is the universal gravitational constant, $M_T(kg)$ is the mass of the planetoid, and each $\{x_k, y_k, z_k\}$ is the spacial coordinate of a discrete, point-like, mass element of the planetoid.

For the sake of explicitness, we can also write the full system in the Cartesian coordinate system.

$$\begin{aligned}
\ddot{x} &= GM_T \sum_{k=0}^{N} \frac{(x_k - x)}{\left((x-x_k)^2 + (y-y_k)^2 + (z-z_k)^2\right)^{3/2}} \\
\ddot{y} &= GM_T \sum_{k=0}^{N} \frac{(y_k - y)}{\left((x-x_k)^2 + (y-y_k)^2 + (z-z_k)^2\right)^{3/2}} \\
\ddot{z} &= GM_T \sum_{k=0}^{N} \frac{(z_k - z)}{\left((x-x_k)^2 + (y-y_k)^2 + (z-z_k)^2\right)^{3/2}}
\end{aligned}$$

II.I Simplifying Assumptions

A number of assumptions can be made which make the problem far more tractable.

- Assume that for the Torus, T, with inner (gauge) radius A, and outer (thickness) radius a, $A >> a$, such that $a/A \approx 0 \Rightarrow a \approx 0$. That is, we assume the torus has no thickness.

- Assume that the density of T is constant.
- Assume that the mass elements of T are stationary – that the torus is not rotating.

II.II Dimensionless Analysis

Another avenue which would certainly be of interest is making the problem dimensionless. As of right now, due to the Universal Gravitational Constant, the scales required to see any sort of noticeable gravitational dynamics are quite high – nondimensionalizing the problem would certainly help with this in terms of reducing computational load, but we do lose some of the intuition for the dynamics at scale.

II.III Numerical Techniques

Now, finding an analytic solution to this problem may prove to be beyond my capabilities – numerical solutions, however, are more within my purview.

To begin, we transform the system of three 2nd order equations to a system of six 1st order equations, by the following substitutions

$$x_1 = x \Rightarrow \dot{x}_1 = \dot{x}$$
$$x_2 = x' \Rightarrow \dot{x}_2 = \ddot{x}$$

Which, substituted into our existing system yields

$$\begin{aligned}
\dot{x}_1 &= GM_T \sum_{k=0}^{N} \frac{(x_k - x_1)}{\left((x_1 - x_k)^2 + (y_1 - y_k)^2 + (z_1 - z_k)^2\right)^{3/2}} \\
\dot{y}_1 &= GM_T \sum_{k=0}^{N} \frac{(y_k - y_1)}{\left((x_1 - x_k)^2 + (y_1 - y_k)^2 + (z_1 - z_k)^2\right)^{3/2}} \\
\dot{z}_1 &= GM_T \sum_{k=0}^{N} \frac{(z_k - z_1)}{\left((x_1 - x_k)^2 + (y_1 - y_k)^2 + (z_1 - z_k)^2\right)^{3/2}} \\
\dot{x}_2 &= x_1 \\
\dot{y}_2 &= y_1 \\
\dot{z}_2 &= z_1
\end{aligned}$$

We see that $\{x_1, y_1, z_1\}$ will be the spacial coordinates of the orbiting body at any given time, whereas $\{x_2, y_2, z_2\}$ will be the velocity. In order to numerically approximate solutions to this system, we give initial conditions $[r_{x,0}, r_{y,0}, r_{z,0}, v_{x,0}, v_{x,0}, v_{x,0}]$

There are a slew of techniques available to approximate numerical solutions to systems of ODEs – I will be using the Python package *scipy.integrate.odeint*.

III. Results

The following are a selection of results taken from my testing of the restricted N-Body problem with a torus configuration. This section is fairly heavy on figures, but I think that's more interesting than just talking about the results in word.

III.I Test Case (N=1)

To start, I just wanted to see if I could replicate the dynamics of a restricted two-body problem – one mass orbiting around a larger, stationary mass (like the earth and the sun). The exact numbers used aren't entirely important, but the dynamics appear to be the elliptical orbit one might expect from such a system – how exciting.

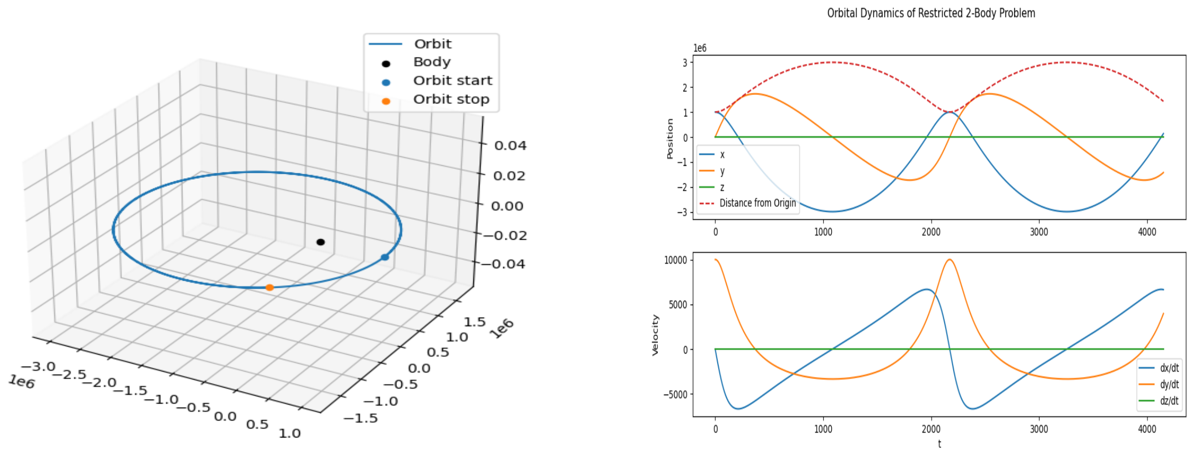

Figure 1: A graphic depiction of an elliptical orbit in the xy-plane of the restricted 2-body problem (left) as well as the position and velocity of the orbiting body as functions of time (right).

III.II Playing Around (N=2,4)

I wanted to play with the system a bit more before jumping into my analysis of the Torus problem, so I tested the dynamics for $N = 2$ and $N = 4$.

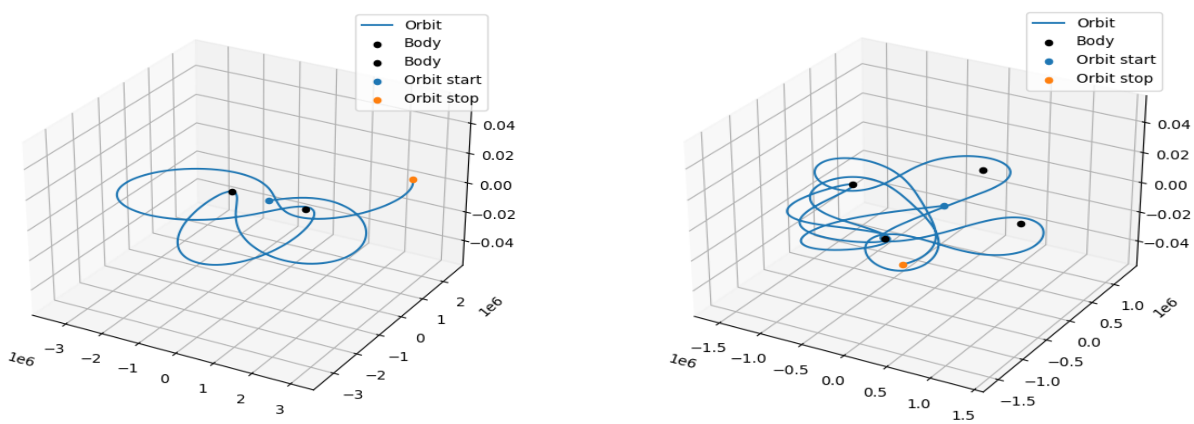

Figure 2: Dynamics of the restricted 3-body (left) and 5-body (right) problems – finding stable orbits for these systems would be an interesting challenge in and of itself.

III.III Semi-Stable Orbits Around T

There are a couple orbits which may be intuited *a priori* and will be (semi)-periodic based on symmetries present in the system.

Due to the fact that I have not yet nondimensionalized the problem – I choose non-significant values for the mass and radius of T as well as the initial conditions. Specifically, $M_T = 1e24kg$, about 1/6th the mass of earth, and $r_T = 1e6m$, about 1/6th the radius of earth.

We first look the initial conditions $x_0 = y_0 = z_0 = v_0 = 0$. Theoretically, these conditions should give us a stable orbit insomuch as the orbiting body does not shift from its initial position and does not gain any velocity. Experimentally, due to the discrete nature of the model, the center of gravity of the system does not lie at the origin, though it is very close, and it will become closer to the true origin as N is increased.

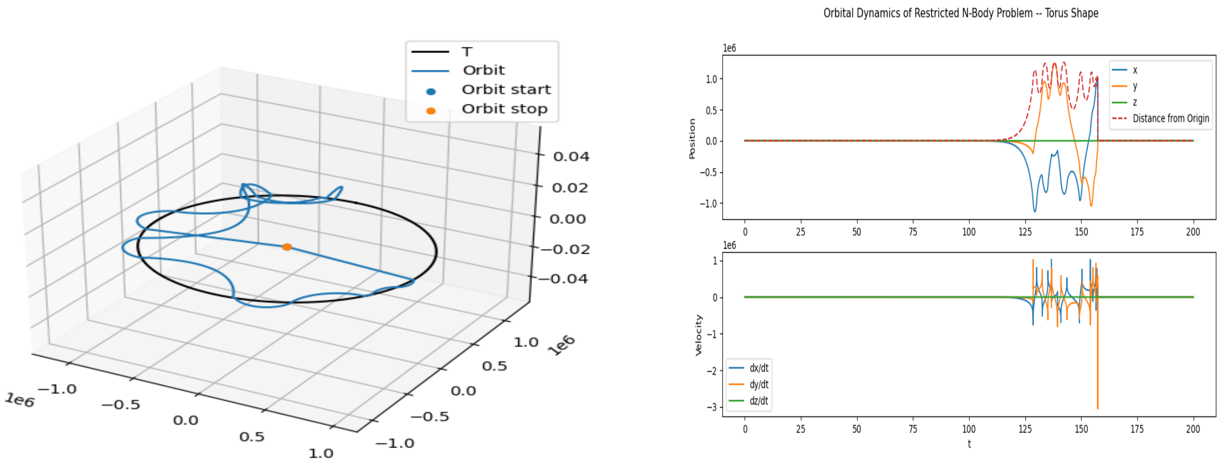

Figure 3: The orbiting body stays close to the origin for some time before small discrete errors cause the body to orbit wildly around T. The sudden jump from orbiting behavior back to the origin is caused by the body colliding with one of the discrete chunks of T, ending the simulation.

Another potential orbit is for the case that the orbiting body is very far away from the torus – in this case, we may treat the torus as a point-mass centered at the origin, and the dynamics will be like those shown in Figure 1. Though these dynamics will hold along any axis – I show dynamics in the $(y = z)$-plane to demonstrate that the dynamics do not rely on the symmetries of the xy-plane.

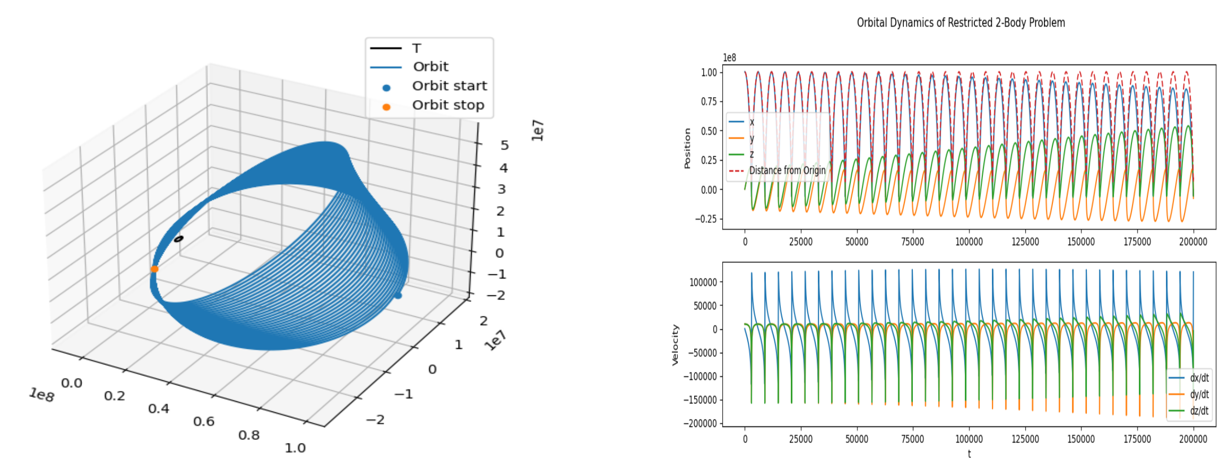

Figure 4: At this scale, T is becomes difficult to see, but is located in the inner left portion of these rings. The magnitude of the orbit remains fairly constant even as the orbital angle undergoes gradual shifts.

Another simplistic orbit is the case that T is positioned in the xy-plane, and the orbiting body is located somewhere along the z-axis with zero velocity components in the x and y directions. These initial conditions lead to sinusoidal oscillations of the orbiting body on the z-axis through the center of T. This and all following simulations are performed with $N = 2500$, which is not high enough to be infallible, but is high enough for the our purposes.

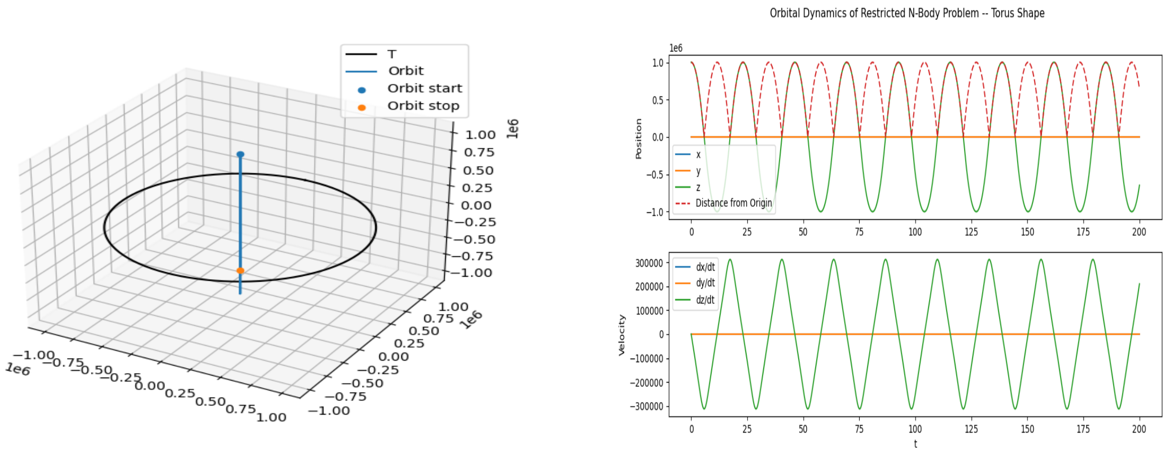

Figure 5: Oscillations in the z-axis due to gravitational effects of T. Though not mathematically rigorous, we can intuit that, if T is continuous, then, the symmetries are such that the orbiting body only experiences a force of attraction in the $\pm\hat{\mathbf{k}}$ direction

Though, as we have seen with our first case, small shifts cause wildly different behaviors in the system over time – let's look at the same setup, but shift the body 100m in the x direction.

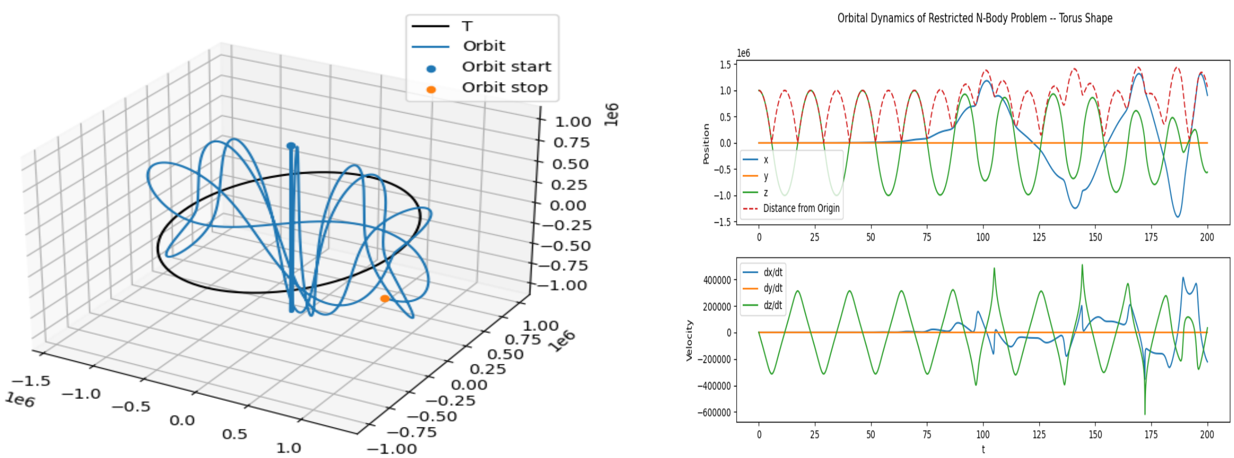

Figure 6: Like Figure 3, we begin with a seemingly stable orbit which mirrors the oscillatory behavior seen in Figure 5, but over time, subtle shifts reveal chaotic behavior of the system.

One more fun example is one in which the initial z and x components of the velocity is offset so that there is more three dimensional behavior.

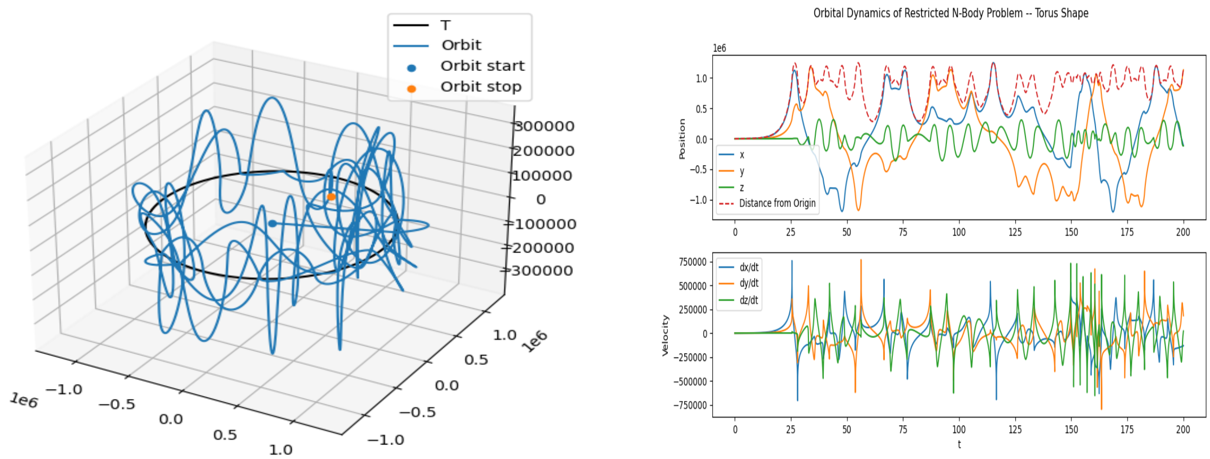

Figure 7: A fun example of the chaotic dynamics – a clearly unstable orbit.

I could honestly play with this simulation for hours just seeing what interesting dynamics would come up.

IV. Conclusions

It is apparent that the gravitational, orbital dynamics about a thin, discretized ring are chaotic. Trivial orbits do exist due to symmetries within the torus system, but, in general, stable orbits are few and far between.

I think it's interesting that, even in the chaotic behavior, we still have the orbiting body staying fairly close to the ring and that it rarely jumps from one end of the ring to the other. I also expected for oscillations within the center of the ring about the center of gravity, but I suppose at such large scales the gravitational effects of masses at one end of the ring are far less than the effects at the other end.

More work is needed to understand the nondimensionalized problem, which I genuinely think would be an interesting area for future projects or just in my own time.

The code is available on my Github (github.com/Ada-Matthews/Restricted_ N_ Body_ Sims), or in Appendix A.

V. Appendix A: Python Code

```
import numpy as np
from matplotlib import pyplot as plt
from scipy.integrate import odeint
pi = np.pi

from matplotlib import cm
from ipywidgets import interact
%matplotlib widget


def odes(X, t, T):
    #constants

    G = -6.67e-11
    mE = 1e24

    #odes
    x1 = X[0]
    y1 = X[1]
    z1 = X[2]
    x2 = X[3]
    y2 = X[4]
    z2 = X[5]

    dx2 = 0
    dy2 = 0
    dz2 = 0

    dx1 = x2
    dy1 = y2
    dz1 = z2

    for i in range(len(T)):
        mag = np.sqrt((T[i][0]-x1)**2+(T[i][1]-y1)**2+(z1)**2)
        dx2 += (x1 - T[i][0])/(mag**3)
        dy2 += (y1 - T[i][1])/(mag**3)
        dz2 += (z1 - T[i][2])/(mag**3)

    dx2 = dx2*mE*G
    dy2 = dy2*mE*G
    dz2 = dz2*mE*G


    return [dx1, dy1, dz1, dx2, dy2, dz2]

## defining the list of stationary masses
r = 8e6
N = 2500
```

```
dtheta = 2*pi/N
thetas = []

for j in range(N):
    thetas.append((j)*dtheta)


T = []
for i in range(len(thetas)):
    T.append([r*np.cos(thetas[i]), r*np.sin(thetas[i]), 0])


#Parameters
tlist = np.linspace(0,2000, 100000) #times at which to be evaluated
IC = [100000,0,0,-10000,-10000,0] #initial conditions


#Evaluate ODEs
x = odeint(odes, IC, tlist, args = (T,))

x1 = x[:,0]
y1 = x[:,1]
z1 = x[:,2]
x2 = x[:,3]
y2 = x[:,4]
z2 = x[:,5]


def graph_orbit(x, tlist):
    fig, [ax0,ax1] = plt.subplots(2,1 ,figsize = (12,6))

    plt.suptitle("Orbital Dynamics of Restricted 2-Body Problem")

    ax0.plot(tlist, x[:,0], label = 'x')
    ax0.plot(tlist, x[:,1], label = 'y')
    ax0.plot(tlist, x[:,2], label = 'z')
    ax0.plot(tlist, (np.sqrt(x[:,0]**2+x[:,1]**2+x[:,2]**2)),
     label = 'Distance from Origin', ls = '--')

    ax0.set_ylabel('Position')
    ax0.legend()

    ax1.plot(tlist, x[:,3], label = 'dx/dt')
    ax1.plot(tlist, x[:,4], label = 'dy/dt')
    ax1.plot(tlist, x[:,5], label = 'dz/dt')
    ax1.legend()

    ax1.set_ylabel('Velocity')
    ax1.set_xlabel('t')
```

```
    return

graph_orbit(x, tlist)
```