Simulation and study of 2 Body Astrodynamics

Dheeraj Vittal Shenoy

Canara First Grade College, Mangaluru

Contents

1 Acknowledgement

I would like to thank **Dhanush Vittal Shenoy**, my brother, for motivating during this project work.I would also like to thank **Luke Smith**, a youtuber, for his tutorials on LaTeX.

2 Abstract

This project focuses on simulating the gravitational two body problem using Python and find few initial conditions for which the orbit of the bodies are stable, in non-inertial frame of reference and in relative frame of reference.

3 Introduction

Gravitational two body problem is a classical mechanics problem of predicting the motion of two massive objects (abstractly viewed as point particles) under gravity. We make use of Newton's law of gravity and numerical methods to find the solution for the orbits of these objects, and simulate their changing positions.

Let m_1 and m_2 be the masses of two massive objects assumed to be point particles, separated by a distance r. We assume that the gravitational interaction is only with these two bodies, and there are no bodies nearby other than the two under consideration.

From Newton's law of gravity, the force F experienced by an object of mass m due to another object of mass M separated by a distance of R is given by,

$$F = \frac{GMm}{R^2} \tag{1}$$

Where G is the Universal gravitational constant, having the value of $6.674 \times 10^{-11}\,\mathrm{Nm}^2/\mathrm{kg}^2$

In vector form, this equation will be

$$\vec{F} = \frac{GMm}{|\vec{R}|^2}\hat{R} \tag{2}$$

Where R is the radius vector from mass M to mass m.

Therefore, the gravitational force between masses m_1 and m_2 is

$$\begin{aligned} F &= \frac{Gm_1m_2}{r^2} \\ \vec{F} &= \frac{Gm_1m_2}{|\vec{r_{12}}|^2}\hat{r_{12}} \end{aligned} \tag{3}$$

where $\vec{r_{12}}$ is the vector from m_1 to m_2, and $\hat{r_{12}}$ is the unit vector in it's direction.

Using Newton's second law of motion, $\vec{F} = m\vec{a}$, we can calculate the acceleration of a body.

Let $\vec{F}_{12}$ be the force on mass m_1 by m_2, then the acceleration of mass m_1 will be,

$$\begin{aligned} \vec{F}_{12} &= m_1\vec{a}_1 \\ \vec{a}_1 &= \frac{\vec{F}_{12}}{m_1} \\ \frac{dv_1}{dt} = \vec{v_1} &= \frac{\vec{F}_{12}}{m_1} \\ \frac{d^2x_1}{dt} = \vec{\ddot{x}_1} &= \frac{\vec{F}_{12}}{m_1} \end{aligned} \tag{4}$$

Integrating equation (4) twice using suitable numerical integration method, we can determine the position $x(t)$ of mass m_1. Similar calculation can be done to find $x(t)$ of m_2.

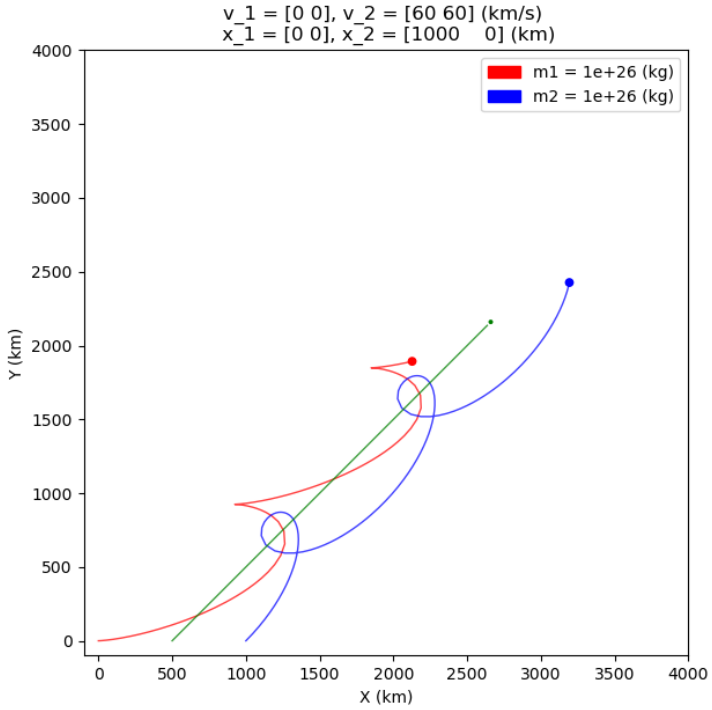
We make use of **algorithm 2.1** mentioned in Curtis (*1*) to compute the motion of two bodies in an inertial frame of reference. We use python programming language to compute and plot the results using the algorithm mentioned above.

The python code for the simulation makes use of **Numpy**, **Scipy** & **Matplotlib** libraries. There are 2 versions of the code, one where the motion of the bodies are with respect to a non-inertial frame of reference, and the other where the motion is relative to one of the body.

4 Results

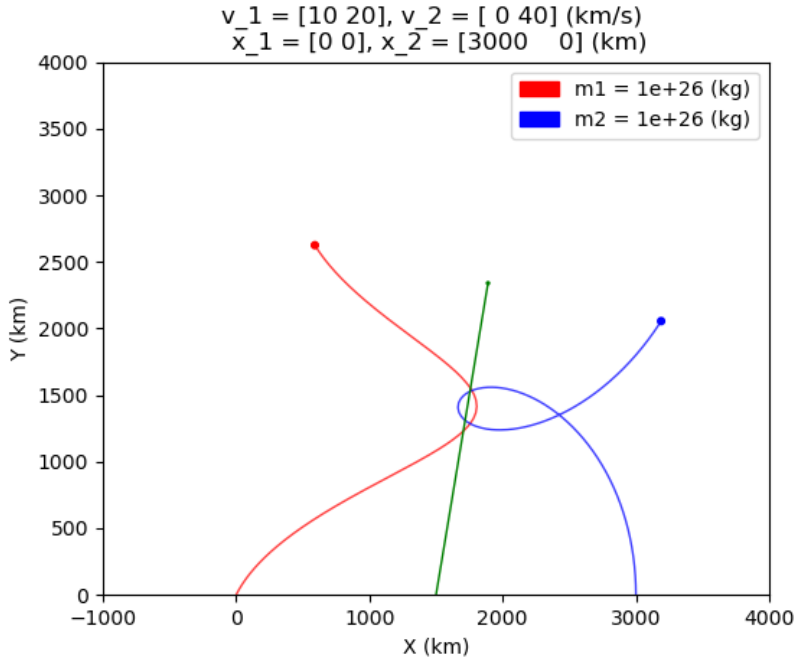
***NOTE*: For the simulation, we use 0.2 hour as the timestep value, if not mentioned**

1. With Position $x_1 = 0\hat{i} + 0\hat{j}$, $x_2 = 1000\hat{i} + 0\hat{j}$ in km and velocity $v_1 = 0\hat{i} + 0\hat{j}$, $v_2 = 60\hat{i} + 60\hat{j}$ in $\mathrm{km/sec}$ and equal masses of mass $1e26$ kg

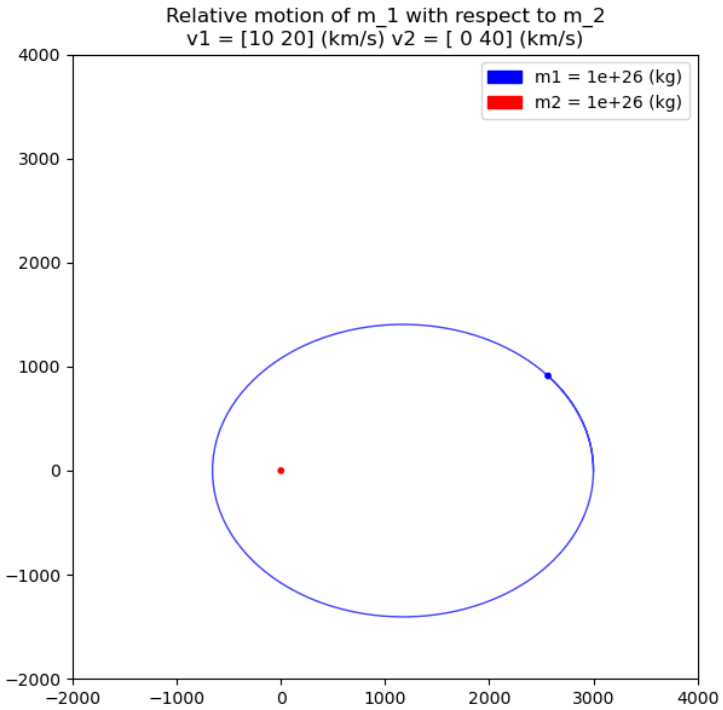


This is the plot of the two body system as seen in a non-inertial frame of reference.
The **green line** represents the center of mass of the system.

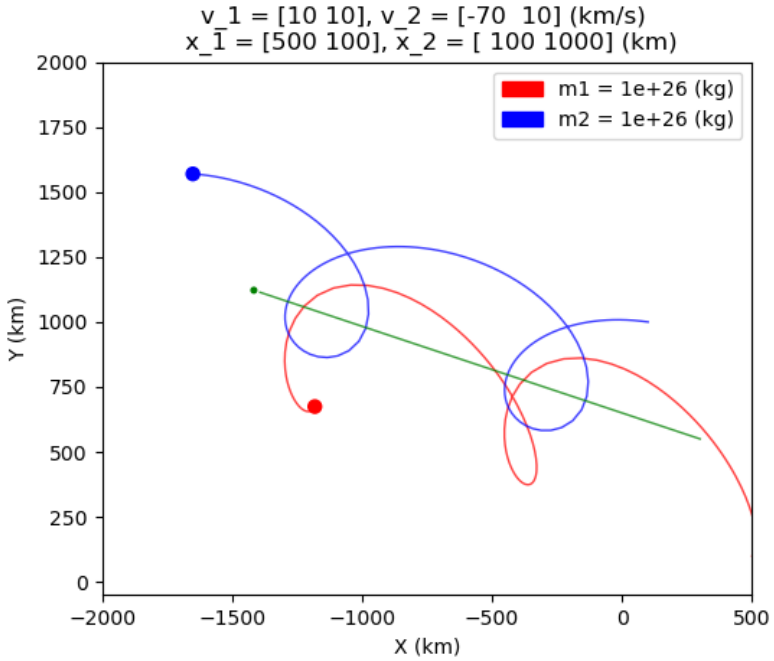
2. With Position $x_1 = 0\hat{i} + 0\hat{j}$, $x_2 = 3000\hat{i} + 0\hat{j}$ in km and velocity $v_1 = 10\hat{i} + 20\hat{j}$, $v_2 = 0\hat{i} + 40\hat{j}$ in $\mathrm{km/sec}$ and equal masses of mass $1e26$ kg



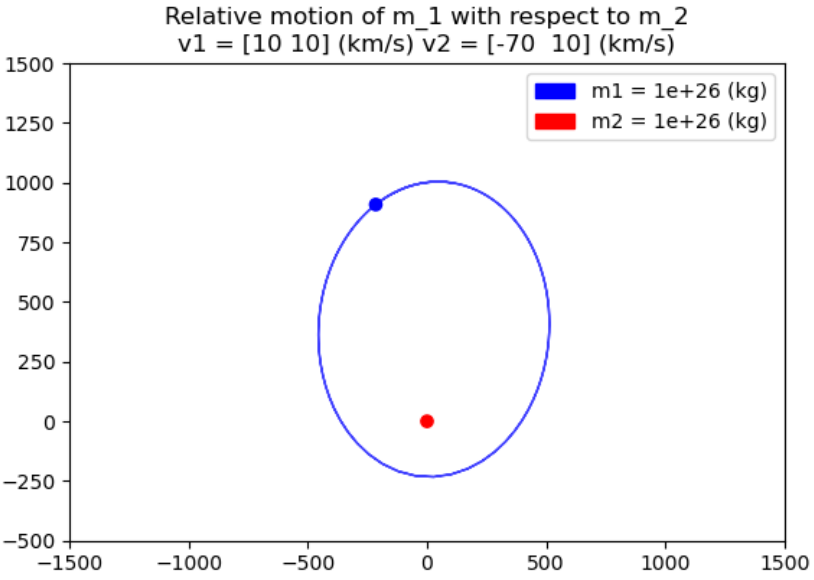
Using the **Algorithm 2.2** mentioned in Curtis (*1*) we can plot the motion of the body relative to another. The below plot shows the relative motion of m_1 with respect to m_2 using the same initial condition.



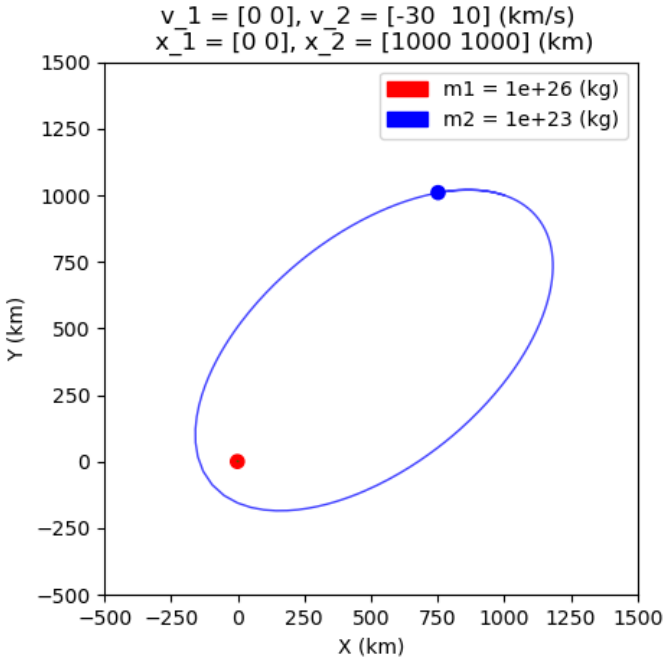
3. With Position $x_1 = 500\hat{i} + 100\hat{j}$, $x_2 = 100\hat{i} + 1000\hat{j}$ in km and velocity $v_1 = 10\hat{i} + 10\hat{j}$, $v_2 = -70\hat{i} + 10\hat{j}$ in $\mathrm{km/sec}$ and equal masses of mass $1e26$ kg



And motion of m_1 relative to m_2,



4. With Position $x_1 = 0\hat{i} + 0\hat{j}$, $x_2 = 1000\hat{i} + 1000\hat{j}$ in km and velocity $v_1 = 0\hat{i} + 0\hat{j}$, $v_2 = -30\hat{i} + 10\hat{j}$ km/sec and masses $m_1 = 1e26$ kg and $m_2 = 1e23$ kg



These conditions allow for a stable orbit of mass m_2 around m_1 shown in the above plot.

5 References

Code

8. D. V. Shenoy, *Two Body Problem Github Repo*, (`https://www.github.com/dheerajshenoy/TwoBodyProblem`).

Books

1. D. Curtis Howard, *Orbital Mechanics for Engineering Students* (Elsevier Butterworth-Heinemann, ed. 4, 2020), ISBN: 978-0-08-202233-0.

3. D. Curtis Howard, *Orbital Mechanics for Engineering Students* (Elsevier Butterworth-Heinemann, 2005), ISBN: 0 7506 6169 0.

4. G. Van Rossum, F. L. Drake, *Python 3 Reference Manual* (CreateSpace, Scotts Valley, CA, 2009).

Articles

2. *Modelling n body problem*, https://www.marksmath.org (`https://www.marksmath.org/classes/Spring2018NumericalAnalysis/code/nBody.html`).

Python Libraries

5. *Matplotlib*, (`https://matplotlib.org`).

6. *Numpy*, (`https://numpy.org`).

7. *Scipy*, (`https://scipy.org`).