Agent of Chaos

Simulating scenarios of the ***N-Body Problem***

by

Aditya Ramdasi

First Year Undergraduate Project for

PHY-1110: **Mathematical Physics I**

Prof. Vikram Vyas

Department of Physics

Ashoka University

Delhi-NCR, India

May 1, 2022

Contents

1 Introduction

1.1 Solar System, Isaac Newton and General Relativity

The **Solar System** is believed to have formed roughly **4.57 billion years ago**, from the gravitational collapse of a huge molecular cloud, composed mainly of hydrogen, helium and small amounts of heavier elements leftover from the death of older stars. The center of this cloud coalesced into the star we call, **the Sun**. The accretion disk of this cloud slowly gave birth to the planets and other bodies orbiting the sun. It created **4 rocky planets -** Mercury, Venus, Earth and Mars; **4 gas giants -** Jupiter, Saturn, Uranus and Neptune, and a bunch of dwarf planets and asteroids.
All these bodies have continued orbiting the sun since the creation of the solar system, due to the gravitational pull of the sun. For the longest part of history, humanity simply didn't know any of these facts.
In **1687**, a brilliant scientist, **Isaac Newton** gave his **''Theory of Universal Gravitation"** building upon the works of numerous other scientists before him, such as Nicholas Copernicus, Johannes Kepler etc. This theory was the first ever theory capable of predicting everything; from why apples fall towards the ground to the motion of planets and stars, to a remarkable degree of precision. It's power mainly came from a completely new and yet consistent system of mathematics which Newton developed and used it in his theory, which is known as *Calculus*.
Newtonian Mechanics and his theory of gravitation offers very powerful tools to study the motions of various cosmic bodies in the universe. We later understood some of its limitations after **Albert Einstein** published his **General Theory of Relativity**. But most of those corrections to Newtonian mechanics often apply in situations more complex than the ones dealt with in this project. I admit, they do apply to certain things like predicting Mercury's orbit around the sun, but for all practical purposes, all the calculations done and simulations created in this project are solely based on Newtonian Mechanics, which gives results quite close to real world observations.

1.2 The N-Body Problem

Isaac Newton published the complete mathematical formalisation of his "Universal Theory of Gravitation" in his book - ***Philosophiæ Naturalis Principia Mathematica***, commonly referred to as - ***Principia***. Apart from describing his theory, Newton also opened up the scientific discussion on a specific problem involving 2 or more than 2 objects with mass, exerting their gravitational influence upon each other. This was the **N-Body Problem**, where "N" represents the number of such massive objects involved in the system. This problem is notorious for being almost impossible to solve analytically, for values of N>2, and has thus troubled physicists, astronomers and even mathematicians for almost 4 centuries now.
Nevertheless, since the development of digital computers in the modern era, we have been able to solve this problem *numerically*, and simulate the interactions for systems with more than 2 bodies on the computer. This project attempts to simulate a few specific cases of this N-body problem, in the context of our Solar System.

2 Aim

This project first establishes the premise of the Solar System being a dynamic system involving multiple bodies with varying masses at varying distances. For all practical purposes, the **center of mass** of the Solar System is considered to be the **center of the Sun** itself, though we know that it isn't the exact case. But considering the relative mass of the sun with respect to all other planets (and their positions too), it is a **perfectly reasonable approximatio**n to make.
We first establish a simple 2-body system, comprising of the Earth revolving around the sun. We then go on to extend this arrangement to 3 other planets of the Solar System - Mars, Jupiter and Saturn. Mercury's orbit is not correctly predicted by Newtonian mechanics, as stated earlier. Thus, this project doesn't simulate Mercury's orbit. Venus hasn't been considered due to its close proximity to both the Sun and Earth, when viewed at an appropriate scale. Uranus and Neptune need very long periods of time to complete their orbit, due to their distance from the sun. This extends the simulation time considerably, along with consuming a lot of processing power.

To sum-up, these 4 planets have been considered because their **relative distribution in masses and position**s cover a wide range of initial conditions. Other bodies (except Mercury) haven't been simulated purely due to **computational constraints.**

We then go on to introduce new hypothetical objects into our toy Solar System. We call these objects, **disrupters**. They have **3 variable initial conditions; point of origin, initial velocity and mass**. We aim to modulate these 3 properties of the disrupters, and see their corresponding effects on our Solar System. Later, we also introduce 2 disrupters simultaneously, and attempt to provide them specific initial conditions such that they cancel each other's gravitational disruptions and leave the Solar System unperturbed.

3 Scenarios simulated

To formalize, the following scenarios are simulated in this project:-

1. A stable 2-body system (Earth and Sun) (2D)
2. A stable 5-body system (Sun, Earth, Mars, Jupiter and Saturn) (2D)
3. A stationary disrupter of 1 solar mass at a specific distance to pull out specific planets of the solar system (2D)
4. A disrupter directed directly towards the sun with some initial velocity (2D)
5. A disrupter directed directly towards the sun with some initial velocity (3D)
6. Disrupters with specific initial conditions to eject each planet separately without perturbing the orbits of other planets (3D for the Earth, rest all 2D)
7. A disrupter approximating ***Proxima Centauri*** with respect to mass (3D)
8. 2 disrupters with zero initial velocity and equal masses which maintain the Solar System's orbital stability (3D)
9. 2 disrupters of equal masses with initial conditions (non-zero velocity) which maintain the Solar System's Stability (2D)

   The code for each of these simulations has been provided in separate python files and should run on any normal computer. This report contains the initial conditions of every scenario, images of the simulation taken at various time instances, and some observations. Interested readers may run the python files on their device to watch the full simulation in real-time.

4 Methodology

4.1 Leapfrog method

The algorithm used in this project for solving the **N-Body problem computationally** is known as the **Leapfrog Method**. It is method of second order integration, which involves finding the velocity after a time interval $\frac{\Delta t}{2}$ instead of Δt, thus increasing the precision. We can write the general formula for distance (x) and velocity (v) using the leapfrog method as follows -

$$\begin{aligned} x_{n+1} &= x_n + \Delta t v_{n+\frac{1}{2}} \\ v_{n+1} &= v_{n+\frac{1}{2}} + a_{n+1}\frac{\Delta t}{2} \end{aligned} \tag{1}$$

This method gives a higher precision than first order methods of numerical integration like the **Euler-Cromer** method, since we are calculating the position after every $\frac{\Delta t}{2}$ time interval, rather than Δt. This prevents the error propagation rate from increasing significantly after every step.

The Leapfrog method even offers **additional advantages** for simulations of Newtonian mechanics, like conserving **angular momentum** of the system, being time-reversible etc.

An even better algorithm to simulate N-Body systems, especially those that turn **chaotic** later, is called the **Runge-Kutta method**. It comes in 2^{nd}, 3^{rd} and 4^{th} order, and offers even more precision. However, it is beyond the scope of this project, but an interesting and important method to learn about and employ in future mechanics simulations.

4.2 Equations of Force and Acceleration

According to Newton's **Universal Theory of Gravitation**, the gravitational force of attraction between objects of mass m_1, m_2 situated at a distance r from each other is given by the expression -

$$F_{12} = \frac{Gm_1m_2}{r^2} \tag{2}$$

It is always attractive in nature, and acts along the line joining the center of masses of the two bodies involved.
The vector form of this equation can be written as follows -

$$F_{ij} = \frac{Gm_im_j}{||r_i - r_j||^2} \cdot \frac{(r_i - r_j)}{||r_i - r_j||} \tag{3}$$

where r_i , r_j are position vectors of the i^{th} and j^{th} body respectively. The initial expression of the force has been multiplied by a unit vector, to specify its direction without changing its magnitude.
From Newton's Second Law of Motion, we can obtain the values of acceleration for each of the bodies, just by dividing equation (3) by the object's mass.

4.3 Conserving Angular Momentum in CoM frame

For running the simulation without violating the laws of physics, we need to conserve angular momentum and energy. Energy gets conserved from our equations of force and acceleration (since gravitational force is known to be a **conservative force**). To conserve angular momentum, we write a small **for loop** in python, which adds up the momenta of all bodies one by one, and equates it to zero. This also gives us the velocity of the sun (due to forces from these N-bodies), since we are considering the Sun as a center of mass of entire Solar System (refer Section -**Aim**).
The mathematical formulation is as follows -

$$\begin{aligned} 0 &= m_sv_s + m_1v_1 + m_2 + v_2 + ... + m_nv_n \\ m_sv_s &= -\sum_{i=1}^{n} m_iv_i \\ v_s &= \frac{-\sum m_iv_i}{m_s} \end{aligned} \tag{4}$$

where m_s and v_s are the mass and velocity of the sun respectively, while m_i and v_i are the masses and velocities of all other bodies in our system.

4.4 Values of Constants Used

The following values of constants were used for simulating all the scenarios listed:-

- Universal Gravitation Constant 'G' = 6.673 $\times 10^{-11}$ $Nkg^{-2}m^2$
- $\pi = 3.141592$
- 1 Astronomical Unit (A.U.) = 149.6 $\times 10^9$ m
- Length of 1 Earth year = 365.25 days

The following table shows the initial conditions used for simulating the stable 5-body Solar System -

Object	Initial Position Vector (AU)	Mass (kg)	Orbital Period (Years)
Sun	(0,0,0)	2.0 x 10^{30}	-
Earth	(1.0 x AU,0,0)	6.0 x 10^{24}	1.00
Mars	(1.5 x AU,0,0)	6.4 x 10^{23}	1.88
Jupiter	(5.2 x AU,0,0)	1.9 x 10^{27}	11.86
Saturn	(9.5 x AU,0,0)	5.7 x 10^{26}	29.46

Table 1: Initial conditions for a stable 5-body Solar System

4.5 Python Library used - VPython

VPython is the Python library used for generating all the simulations used in this project. It has many useful inbuilt functions such as *sphere()* to create a spherical object, *b.velocity* to assign the value of velocity to object "b" etc. It is also very user-friendly, has an easy syntax and requires no additional installations.

5 Simulations and Observations

The colour scheme followed in the simulations is as follows -

- Yellow - Sun
- Blue - Earth
- Red - Mars
- White - Jupiter
- Green - Saturn
- Purple - Disrupter(s)

5.1 2-body system

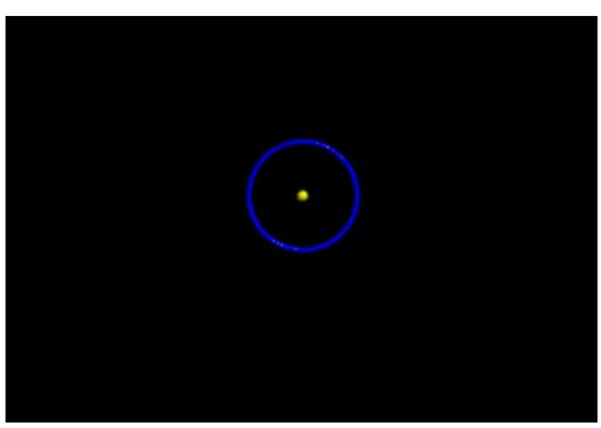


Figure 1: Earth revolving around the sun

Figure 1 shows the most basic configuration of all, a simple 2-body system. It is a stable, 2D system.

5.2 5-Body Solar System

(a) Start of motion (b) Completed orbits

Figure 2: 5-Body Stable Solar System

Figure 2(a) shows the start of our simulation, while 2(b) shows the the completed orbits of all 4 planets. The initial conditions for this are as given in Table 1.

5.3 Stationary disrupter

In this scenario we try to **calculate the initial condition**s for a **stationary disrupter**, to pull out 1 or more than 1 planets **out** of the Sun's sphere of influence.
Let's assume that a cosmic body of mass m_d suddenly **popped into existence** at a distance r_d from the sun. In the following mathematical derivation, we find a relation between m_d and r_d such that a specific planet (with mass m_i and radius of orbit r_i) is pulled out from orbiting the sun.
*Note that we are neglecting the gravitational effects of other planets on this specific planet, since their masses are too small compared to the sun and the disrupter.
Consider the instance of time when the sun, the planet and the disrupter are along the **same line**. The net-force on the planet due to the sun and the disrupter is:

$$F_{planet} = -\frac{Gm_s m_i}{(r_i)^2} + \frac{Gm_d m_i}{(r_d - r_i)^2}$$

For the boundary condition, we equate this to zero, giving us -

$$\frac{m_s}{(r_i)^2} = \frac{m_d}{(r_d - r_i)^2}$$

On further simplification and taking square roots on both sides, we get -

$$\boxed{\frac{r_d}{r_i} = 1 + \sqrt{\frac{m_d}{m_s}}} \tag{5}$$

For simulating a concrete example here, let $m_d = m_s$, and let $i = 4$ i.e. planet to be ejected is Saturn. On substituting the required value of r_i for Saturn, we get $r_d = 2r_i = 19$AU. If we consider the scenario when the disrupter is along $y = x$ and Saturn starts at (9.5AU,0,0), then we need the disrupter to be at a distance of $\frac{19}{\sqrt{2}} = 13.4$AU, for it to escape the Sun's influence.
However, after simulating it was understood that since the disrupter itself is not stationary and starts accelerating towards the sun from t=0, the value of r_d calculated from the expression above is **inaccurate.** There was an **additional factor of approximately** $\sqrt{2}$ that needs to be multiplied to the r_d value to get the limiting condition distance. This changing factor for increasing r_d which occurs due to the acceleration of the disrupter towards the sun, was later understood to **depend on the initial configuration** of Saturn as well as the disrupter.
Figure 3 shows how Saturn goes **tangentially** after ***just* escaping** the Sun's gravitational influence. Here $r_d = 13.4\text{x}\sqrt{2} = 19$AU.

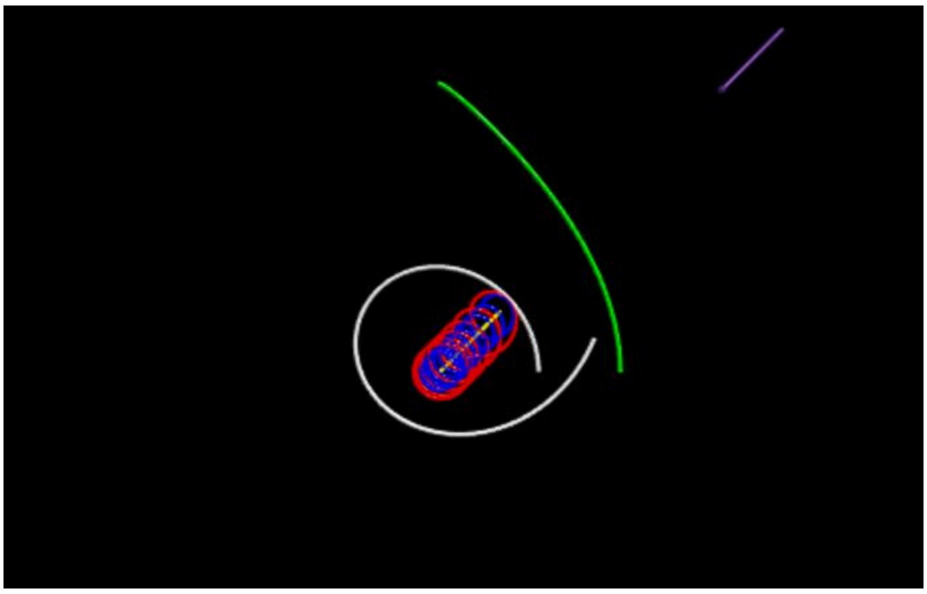


Figure 3: Modified limiting distance for ejecting Saturn

5.4 Disrupter towards sun (2D)

This simulation involves projecting a disrupter towards the sun with a **certain initial velocity**. Since we are not modelling the impact of collisions in this project, **2 opposing case**s for the direction of the disrupter's final velocity are observed; either it **goes straight** through or it **turns back**.
For this scenario, we first derive a critical threshold limit on the disrupter's initial velocity mathematically. We then implement it on Python to verify the threshold, and note down exceptions (if any).
Theoretical derivation for the threshold value of velocity - Conserving the total energy of the 2-body system, we have

$$\text{Gravitational P.E.} + \text{Initial K.E.(disrupter)} = \text{Final K.E.(disrupter)} + \text{Final K.E.(sun)}$$

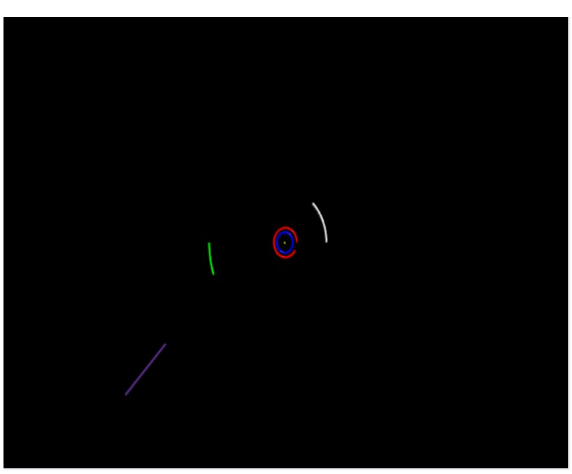
$$-\frac{Gm_s m_d}{r} + \frac{1}{2}m_d(u_d)^2 = \frac{1}{2}(m_d(v_d)^2 + m_s(v_s)^2)$$

$$-\frac{2Gm_s m_d}{r} = m_d(v_d^2 - u_d^2) + m_s v_s^2$$

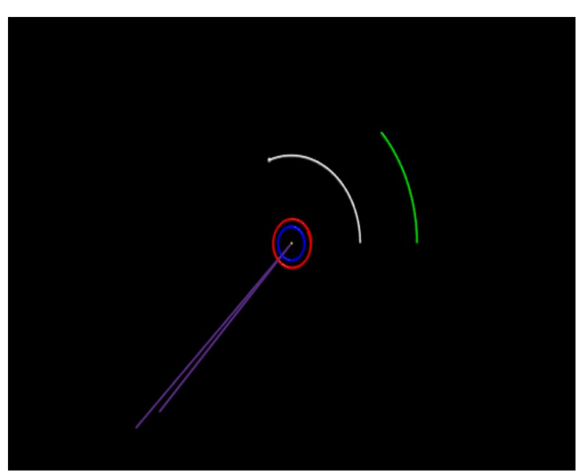
For limiting condition, $v_s \to 0^+$ and $v_d \to 0^-$. We put both these values as 0 to get -

$$u_d^2 = \frac{2Gm_s}{r} \tag{6}$$

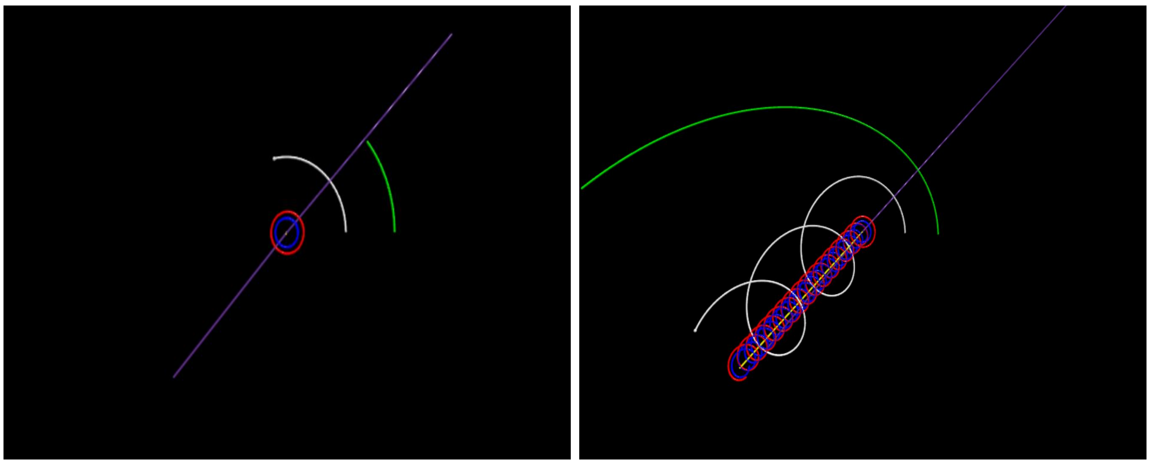
Here, since we are starting our disrupters from (-10AU,-10AU,0), we put the value of $r = \sqrt{2}$, and calculate the value of u_d to be roughly **11.23 km/sec**. Verification by simulation (m_d = 9 x 10^{27} kg) -



(a) Initial positions



(b) Disrupter goes back just below threshold velocity, v=11.2km/s



(c) Disrupter goes straight just above threshold velocity, v=11.3km/s

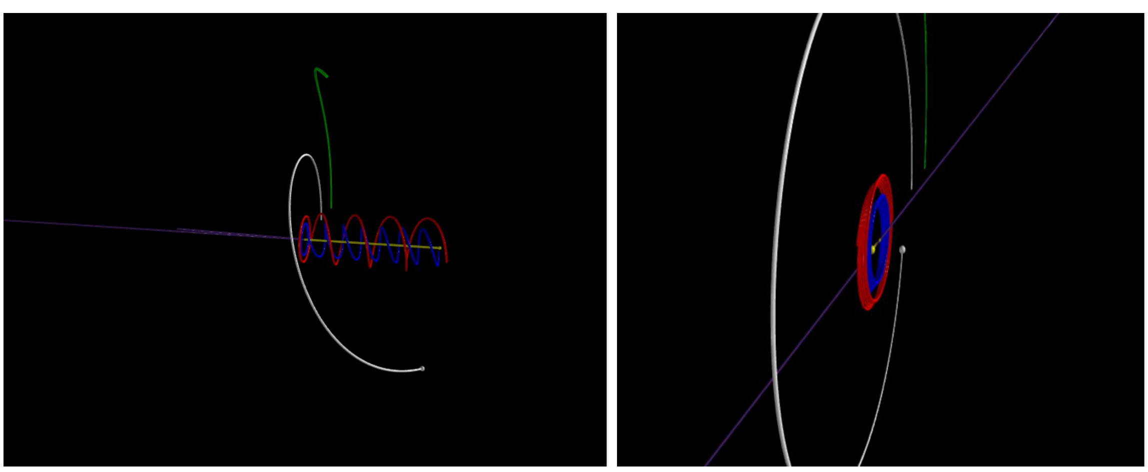
(d) Goes straight well above threshold velocity, v=15km/s

Figure 4: Demonstrating threshold velocity of approach for disrupter

5.5 Disrupter towards sun (3D)

Initial conditions -

- m_d = 9 x 10^{27} kg
- u_d = 9000 m/sec
- starting position of disrupter for (a) = (0,0,-10AU)
- starting position of disrupter for (a) = (-10AU,-10AU,-10AU)



(a) Disrupter along z axis

(b) Disrupter along tilted axis

Figure 5: Disrupter in 3 dimensions

We can see that even though the disrupter gives a linear acceleration the the sun, the orbital stability of the planets remains intact in both these cases, and the trajectories form a helical structure.

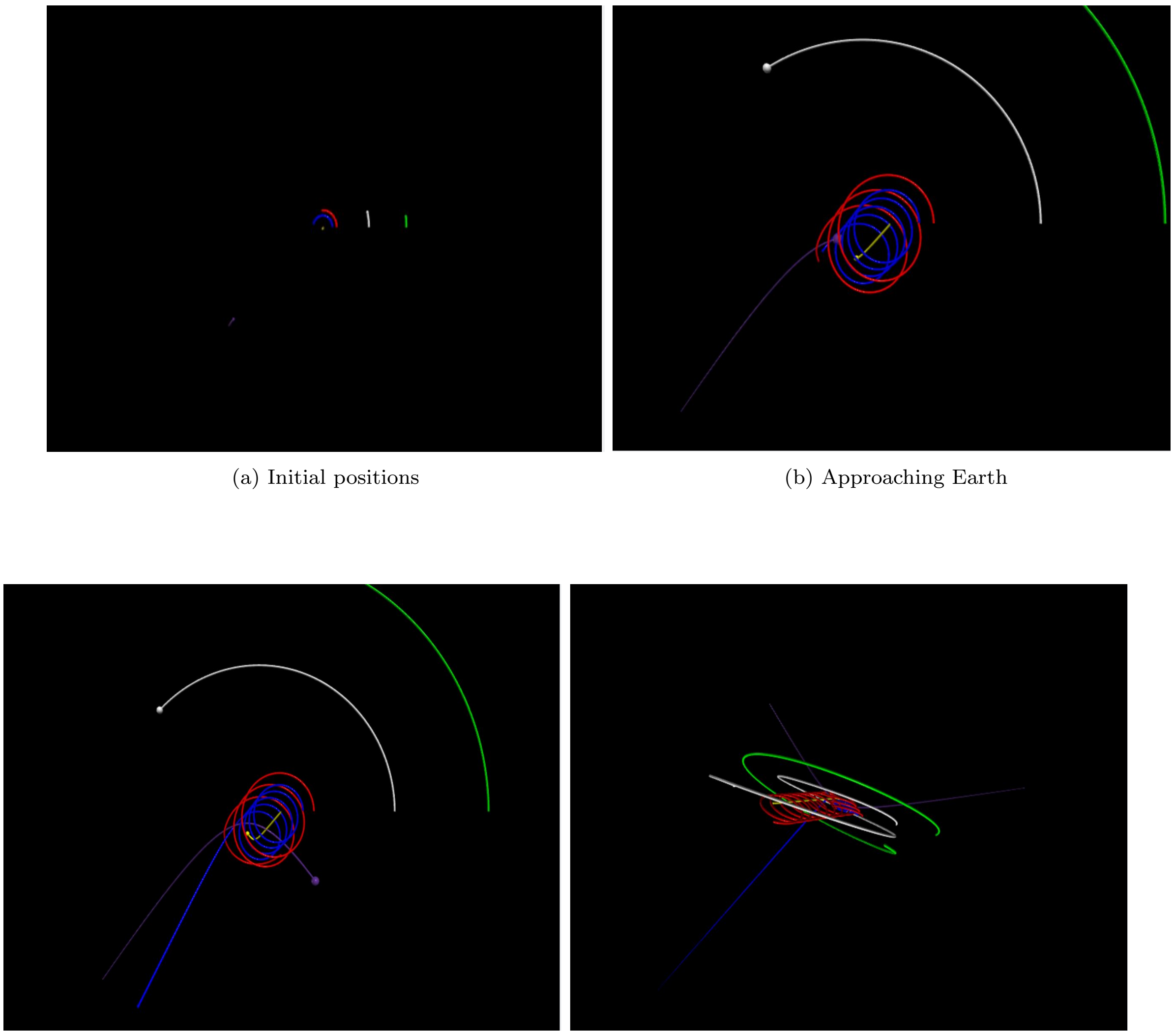
5.6 Disrupters ejecting specific planets

5.6.1 Ejecting Earth (3D)

Initial conditions -

- m_d = 2 x 10^{29} kg
- u_d = 10000 m/sec

- Initial position of disrupter = (-15AU, -12AU, -13AU)



(a) Initial positions (b) Approaching Earth

(c) Earth ejected (d) 3D view

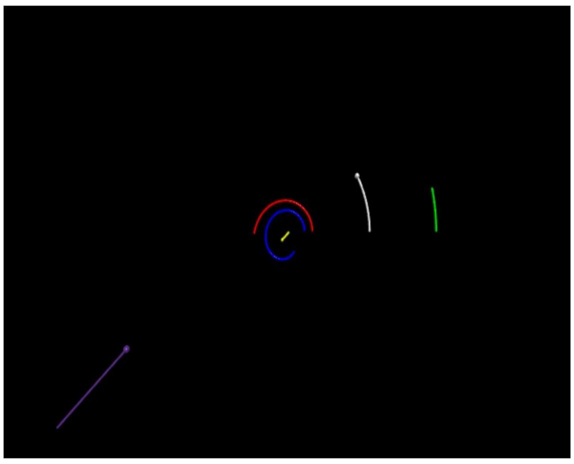
Figure 6: Disrupter ejecting Earth

We can see that from these initial conditions on Earth is ejected, while the sun travels linearly and the orbital stability of other planets is intact, giving us a **helical trajectory**.

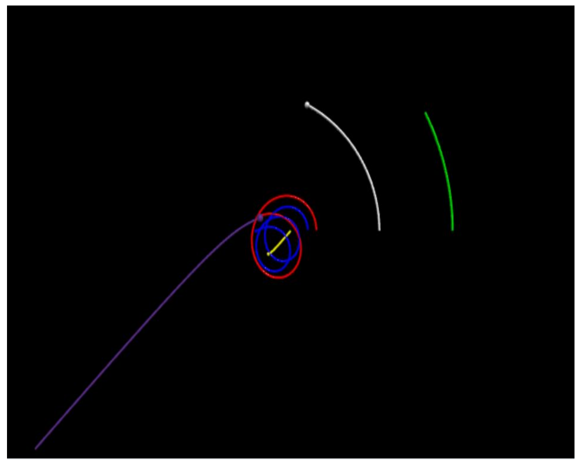
5.6.2 Ejecting Mars(2D)

Initial conditions -

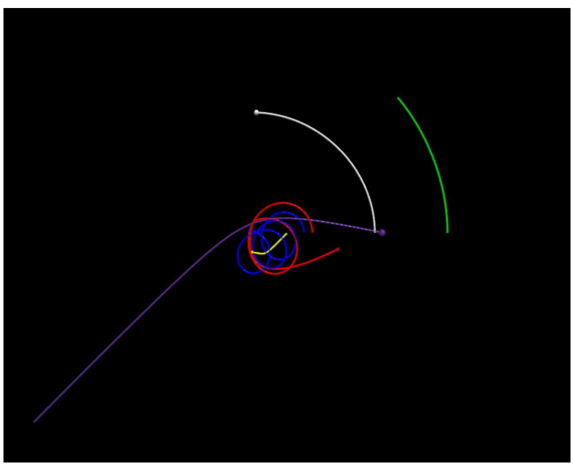
- $m_d = 2 \text{ x } 10^{29}$ kg
- $u_d = 19000$ m/sec
- Initial position of disrupter = (-15AU, -10AU, 0)



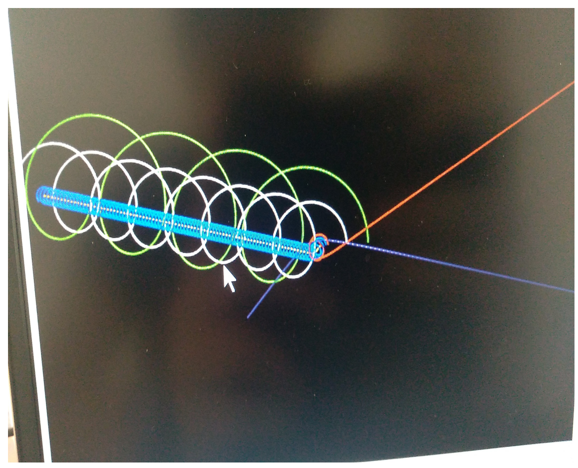
(a) Initial positions



(b) Approaching Mars



(c) Mars Ejected



(d) Other planets continue in a stable orbit

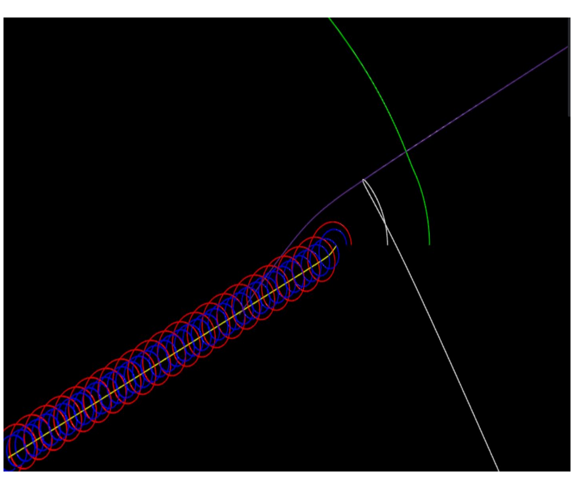
Figure 7: Disrupter ejecting Mars

We can see that from these initial conditions only Mars is ejected, while the sun travels linearly and the orbital stability of other planets is intact, giving us a **cycloidal trajectory**.

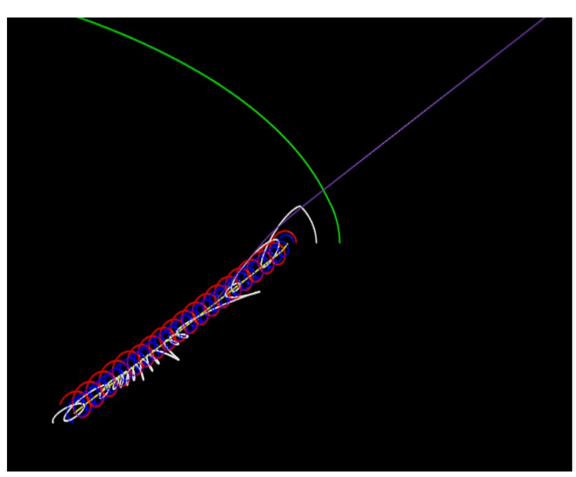
5.6.3 Ejecting Jupiter and Saturn(2D)

Initial conditions -

- $m_d = 2 \text{ x } 10^{29}$ kg
- (a)$u_d = 25000$ m/sec, (b)$u_d = 31000$ m/sec,
- Initial position of disrupter; (a) (-10AU,-5AU,0) (b)(15AU,-10AU,0)



(a) Saturn and Jupiter cleanly ejected



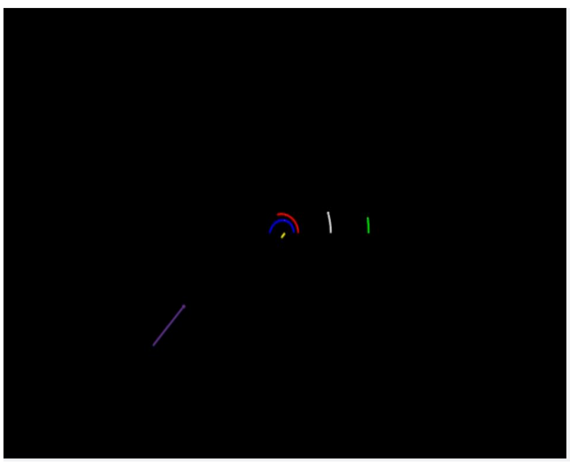
(b) Jupiter's chaotic orbit

We can see that in the first scenario, Jupiter is cleanly ejected along with Saturn. However, in the second scenario, Saturn is cleanly ejected while Jupiter is pushed into a wild, chaotic orbit around the Sun.

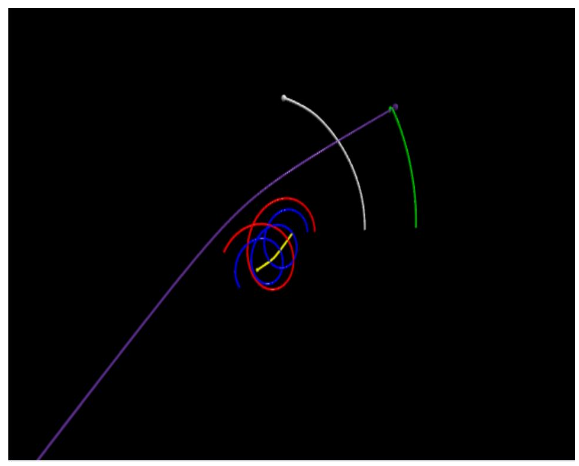
5.6.4 Ejecting Saturn (2D)

Initial conditions -

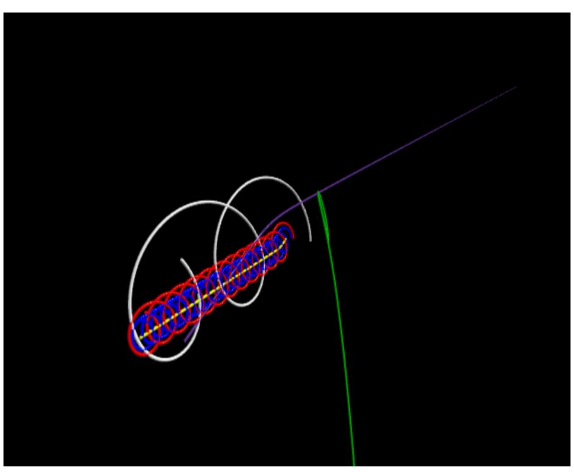
- $m_d = 2 \text{ x } 10^{29}$ kg
- $u_d = 25000$ m/sec
- Initial position of disrupter (-15AU,-10AU,0)



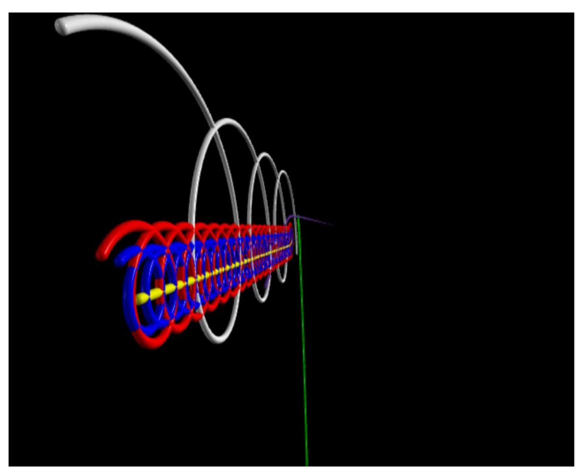
(c) Initial positions



(d) Contact with Saturn



(e) Saturn Ejected, rest continue in stable orbits



(f) 3D zoomed in view of trajectories

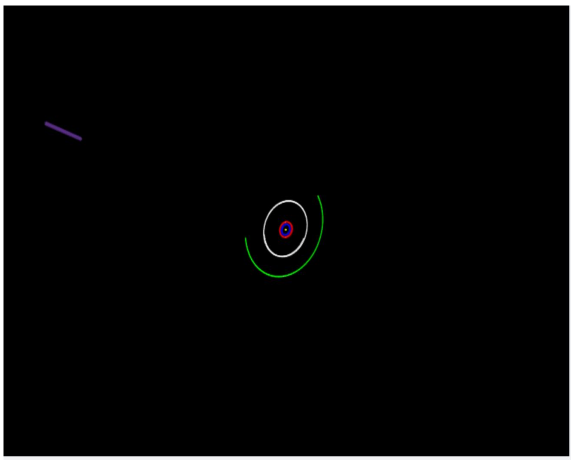
Figure 8: Disrupter ejecting Saturn

From the figure we can see the Saturn is cleanly ejected by this disrupter, while the orbital stability of the rest of the solar system is maintained, with planets following a **cycloidal trajectory**.

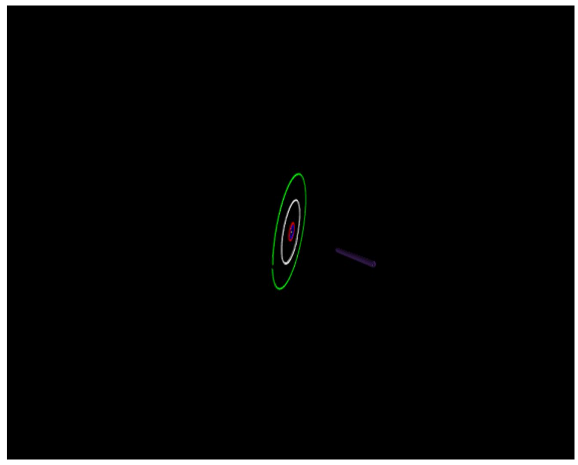
5.7 Ejecting all 4 planets with Proxima Centauri

Initial conditions -

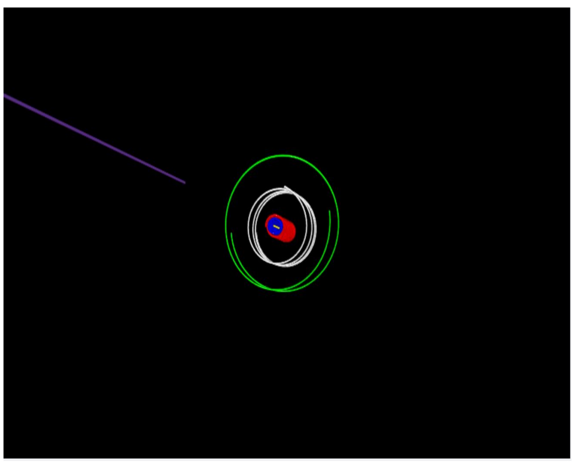
- $m_d = 2.45 \text{ x } 10^{29}$ kg (mass of Proxima Centauri)
- $u_d = 0$ m/sec, we let it accelerate naturally towards the sun
- Initial position of disrupter (-30AU,25AU,20AU)



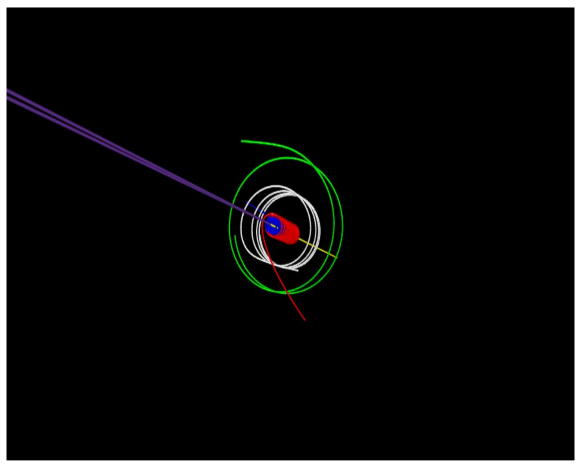
(a) Initial positions 2D view



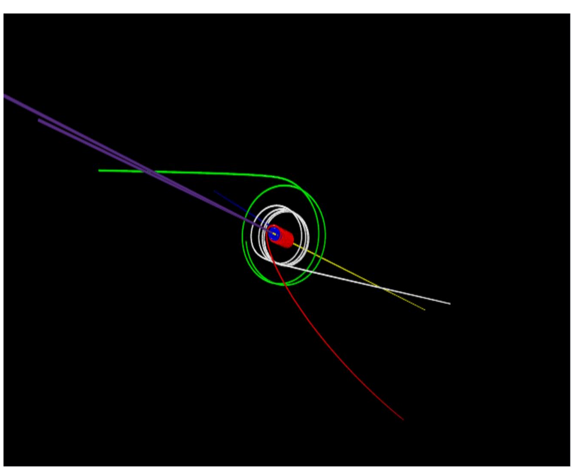
(b) Initial positions 3D view



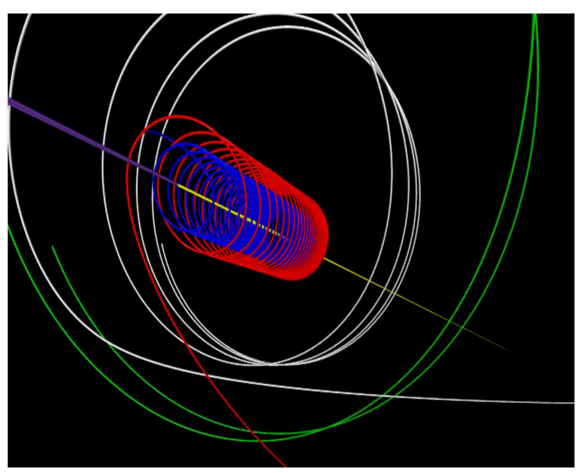
(c) Effects of Proxima Centauri on Solar System before collision



(d) Contact



(e) All 4 planets ejected



(f) Zoomed in view of helical trajectory

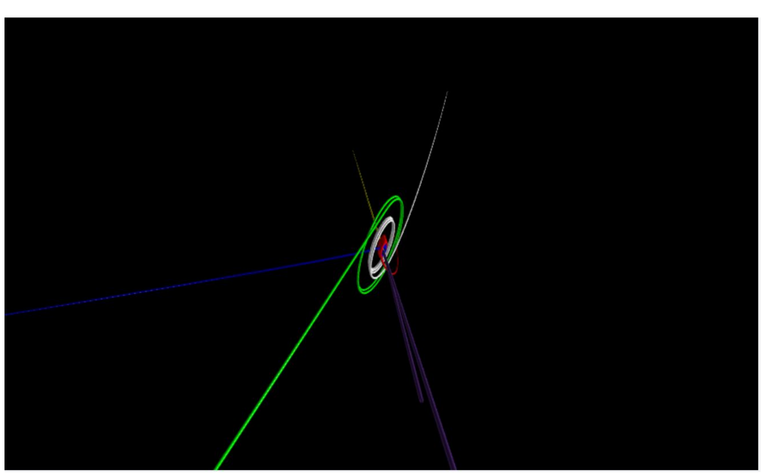


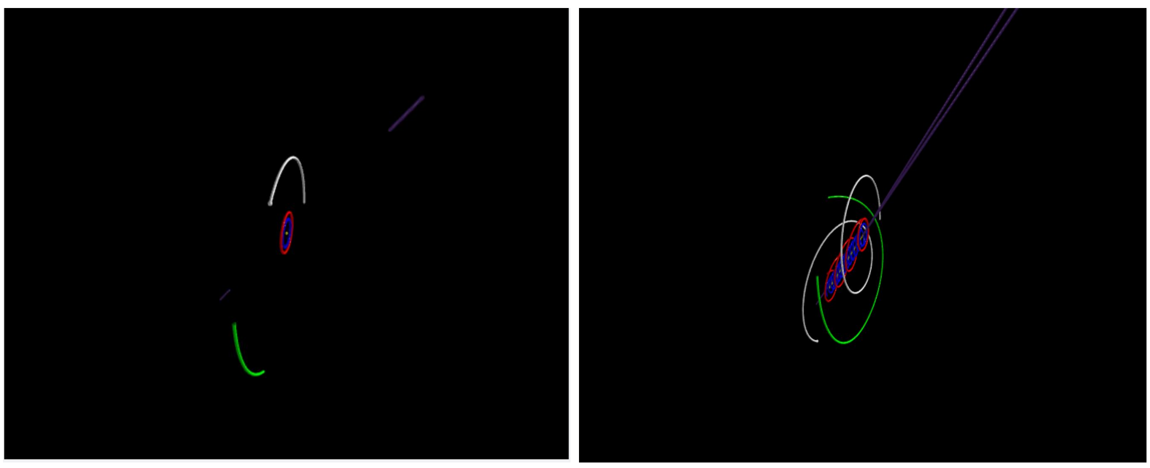
Figure 9: All 4 planets ejected in different directions, 3D view

In this simulation, we used a disrupter of the mass of Proxima Centauri. In the real world, Proxima Centauri is much further away to have a tangible gravitational effect on the Solar System. However, for ejecting all 4 planets out of their orbit, the disrupter resembling Proxima Centauri was released from a certain point in space, as specified above.

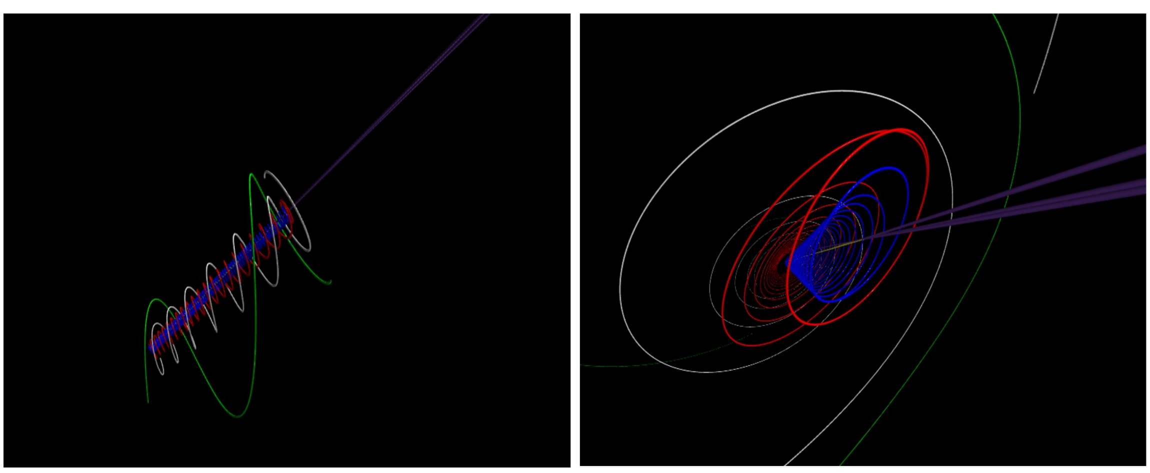
5.8 2 Disrupters starting from rest(in 3D) - maintaining a stable Solar System

Initial conditions -

- $m_d = 9 \text{ x } 10^{27}$ kg for both
- $u_d = 0$ m/sec for both, we let the disrupters accelerate towards the sun from rest
- Initial position of disrupters (3AU,-10AU,-8AU) and (-3AU,10AU,8AU)



(a) Initial positions 3D view (b) Contact and helical trajectories



(c) View from the top (d) Zoomed in view along axis of the helix

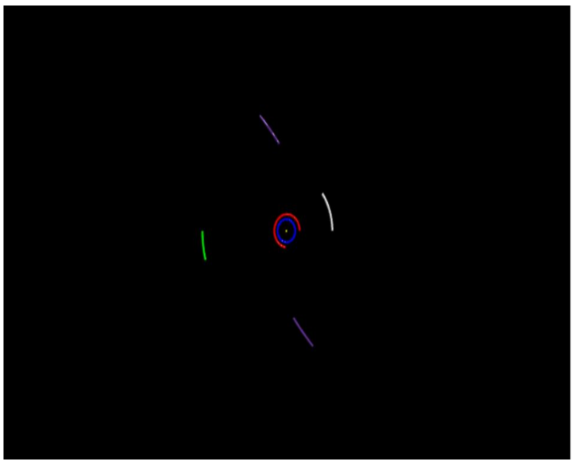
Figure 10: 2 symmetric disrupters in 3D

We can clearly see the beautiful helical structures made by the trajectories of the planets revolving around the sun in the images above. The orbital stability is still intact.

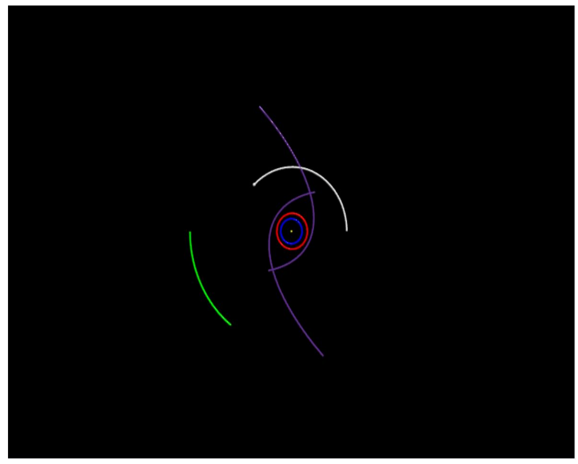
5.9 2 Disrupters with initial velocity(2D) - maintaining a stable Solar System

Initial conditions -

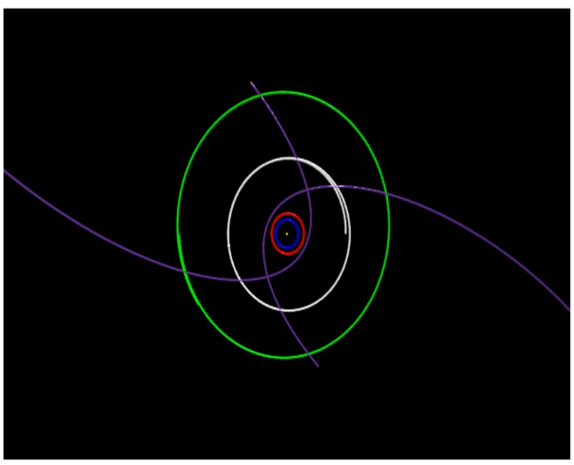
- $m_d = 9 \text{ x } 10^{27}$ kg for both
- $u_d = 8000$ m/sec for both
- Initial position of disrupters (3AU,-10AU,0) and (-3AU,10AU,0)



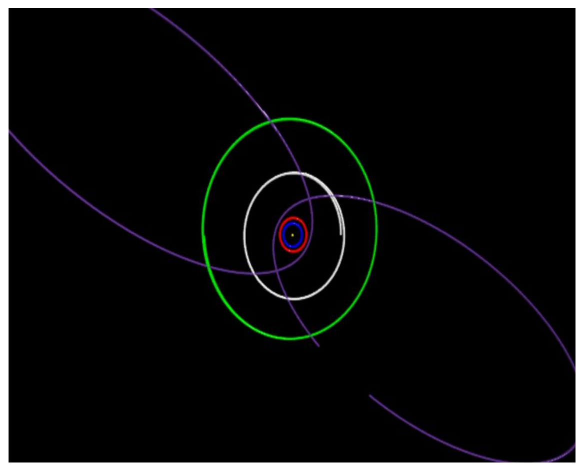
(a) Initial positions



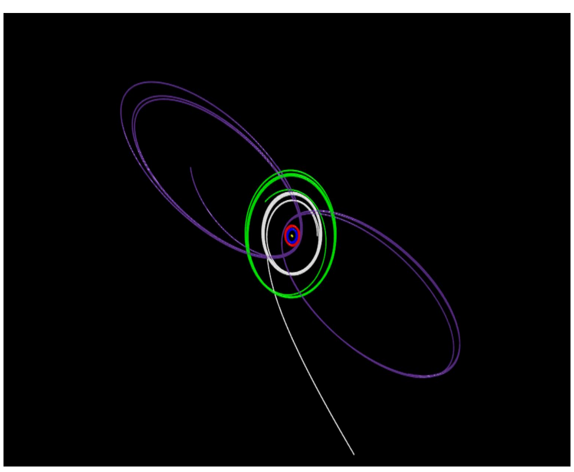
(b) Went symmetrically around the sun



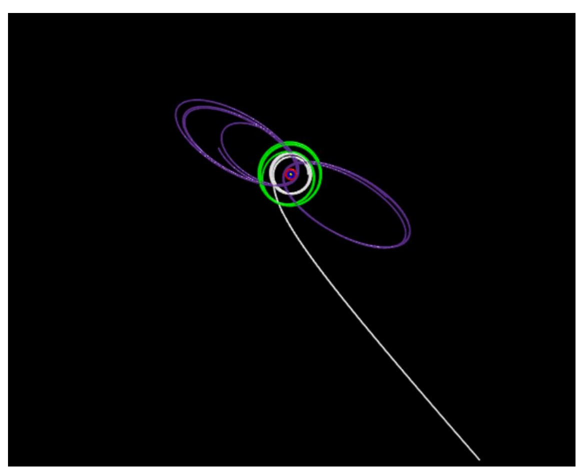
(c) Disrupters enter into elliptical orbits around sun



(d) 1 disrupter completes an orbit



(e) Precession after a few complete orbits



(f) Stable precession but Jupiter Ejected

Figure 11: 2 symmetric disrupters in 2D

We can see that the disrupters enter into a symmetrical, elliptical orbit around the sun without hampering the solar system's stability. After a lot of time i.e. 4-5 complete elliptical orbits around the sun, significant precession is observed in the planet trajectories, and later even Jupiter is ejected out of the Solar System.

6 Limitations

Some of the limitations of the simulations built and calculations done in this project are:

1. **No measurement of time**: Since there was no easy way to measure the passage of time for various events to take place in the simulations, this project can't comment anything on the time taken for certain instances to happen in these simulations.

2. **Usage of Leapfrog**: As most N-body systems turn chaotic after a certain time, Leapfrog method may not always necessarily converge.

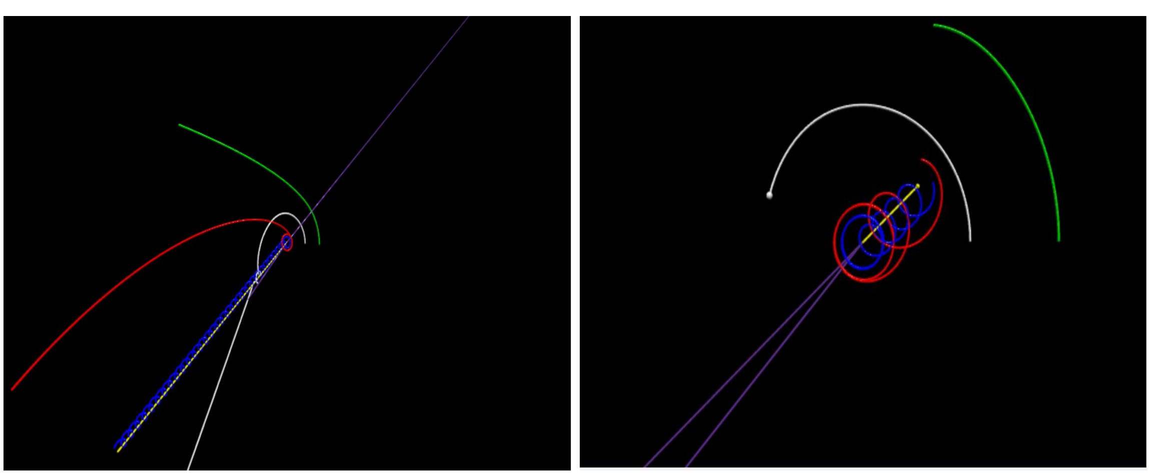
3. **Energy and Angular Momentum conservation**: As mentioned previously in the limitations of the leapfrog method, there is no check in place for ensuring that the total energy of the system remains constant throughout the simulation. A discussion with a senior informs me that when using leapfrog, the total energy does oscillate about a certain value, which propagates an error in the simulation as more time passes. Coding a function to calculate the total energy of the system, calling it at regular intervals and plotting the variation of energy with time can tell us the optimum length of time intervals for least deviation and error propagation. A similar function can also be written for checking the conservation of angular momentum in the system. This is one of the most important actionable identified for the extending the scope of this project.

4. **Sizes and shape of objects**: The radii of the cosmic bodies simulated are not their real values. Moreover, no considerations have been made for factoring in the actual shape of these bodies.

5. **Collisions**: There is no code written in this project for factoring in the changes that will occur due to collision of different cosmic bodies.

7 Exceptions

An interesting exception was observed during running trials for **scenario 5.4**. While the threshold limit for calculated to be **11.23km/sec** and verified by the simulations, some particular values for velocity were found to not obey this result. These were **13 km/s, 13.10 km/s, 13.11 km/s and 10 km/s**. I haven't found an explanation yet for this exceptional behavior, and I thus keep it open as an investigation necessary for further research. Images for 2 of these cases can be found below -



(a) v=10 km/sec, still goes straight through

(b) v=13 km/sec, still turned back

8 Conclusion

The basics of solving the N-Body problem using the Leapfrog method of numerical integration were understood. Various simulations of disrupters with different initial conditions were made to help gain an intuitive sense of how multi-body dynamic systems progress with time. Finally, limitations to be corrected and exceptions to be explained in future projects were also discovered in this process.