Report on

TRAJECTORIES IN THE UNIVERSE

submitted by

Jemish Naliyapara (18114)
Ketan Jadav (18111)
Aayush Mishra (18005)
Ravi Charan (18195)

under the guidance of

Dr.SUNIL PRATAP SINGH
Assistant Professor
Department of Physics

INDIAN INSTITUTE OF SCIENCE EDUCATION AND RESEARCH BHOPAL
APRIL 2021

DECLARATION

We hereby declare that the material, which we now submit for assessment under the report title "TRAJECTORIES IN THE UNIVERSE ", is entirely our own work and has not been taken from the work of others except to the extent that such work has been cited and acknowledged within the text of this report.

IISER Bhopal	Jemish Naliyapara
April 2021	Ketan Jadav
	Aayush Mishra
	Ravi Charan

ACKNOWLEDGMENT

We are grateful to Dr. Sunil Pratap Singh for providing us an opportunity of a project which gave us a platform to try and use the skills learnt in our academic curriculum for some original work. We also thank him for his continuous guidance throughout the development of this project.
We would also like to thank Dr. Sebastian Wuster for teaching us Classical Mechanics from where we got the idea for this project.

Jemish Naliyapara
Ketan Jadav
Aayush Mishra
Ravi Charan

Contents

1 Abstract

Our aim in this project is to study the dynamics of the objects in the universe considering the gravitational interactions among them. We mainly work on getting trajectories of various system of bodies by solving differential equations obtained from the expressions of their gravitational forces. We consider three scenarios for the same in this project. In one body problem, we obtain the trajectory of the earth, keeping the sun stationary. Similarly, we work on a two-body problem and get the orbits of both bodies in different initial conditions. We then proceed to derive the equations of motion for three-body problem and using these equations we find the trajectories in various situations.

We analyze the motion by numerically solving the differential equations of motion using the Runge-Kutta (RK4) method with pre-determined initial conditions. We also analyze the motion in different frames of references to get a better physical insight.

2 Introduction

Newton's law of gravitation states that every object in the universe exerts an attractive force on every other object by virtue of its mass that is directly proportional to the product of their masses and inversely proportional to the square of the distance between their centers.

$$F_g = \frac{Gm_1m_2}{r^2}$$

where m_1 and m_2 are the masses of object under consideration and r is the distance between their centers and G is the universal gravitational constant, has the value $G = 6.6742 * (10^{-11})(m^3/kgs^2)$.

According to Newton's second law of motion the acceleration of an object is directly proportional to the net force acting on it and the direction of acceleration is in direction of the net force.

$$\vec{F}_{net} = m\vec{a}$$

where m is the mass of the object and $\vec{a}$ is the acceleration.

Newton's second law of motion as applied to a body m_1 is $F_{12} = m_1\ddot{R}_1$, where $\ddot{R}_1$ is the absolute acceleration of m_1. Combining this with Newton's law of gravitation yields

$$m_1\ddot{R}_1 = \frac{Gm_1m_2}{r^2}\hat{u}_r$$

Likewise,

$$m_2\ddot{R}_2 = -\frac{Gm_1m_2}{r^2}\hat{u}_r$$

For further insight on above equations refer to sections 4.1 for two body problem and 5.1 for three body problem.

We solve these coupled ordinary differential equations to find the trajectories of the bodies in consideration using numerical integration method Runge-Kutta 4 (RK4) (refer to appendix). We assume the initial conditions the system is in and write down the program for RK-4 method using Python programming language. After multiple iterations, we were able to write an efficient and compact program for the same which can be found in appendix. We then plot the obtained results in various frames of references and draw inferences. We also compare the obtained trajectories with the known trajectories.

3 One Body Problem

In One Body system we deal with system having one body stationary and study the dynamics of the other body in stationary mass frame without any external force.

3.1 Theory Of One Body System Dynamics

We solve one body system in frame of one mass assuming it to be stationary and make it origin of coordinate system. We solve it in polar coordinates.
Let the mass in motion has initial velocity v_i and at distance from origin r_i so its Lagrangian is given by

$$\mathcal{L} = T - V = \frac{1}{2}m(\dot{r}^2 + r^2\dot{\phi}^2) - (-\frac{GMm}{r})$$

Using Euler-Lagrange equation

$$\frac{\partial \mathcal{L}}{\partial \phi} = \frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{\phi}}$$

$$m(2r\dot{\phi}\dot{r} + r^2\ddot{\phi}) = 0$$

$$\ddot{\phi} = \frac{-2\dot{r}\dot{\phi}}{r} \qquad \text{(......eq.1)}$$

In other words

$$mr^2\dot{\phi} = constant = L \qquad \text{(......eq.2)}$$

where L is angular momentum of the system with respect to origin.

$$\frac{\partial \mathcal{L}}{\partial r} = \frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{r}}$$

$$m\ddot{r} = -\frac{GMm}{r^2} + mr^2\dot{\phi}$$

$$\ddot{r} = -\frac{GM}{r^2} + r\dot{\phi}^2 \qquad \text{(.....eq.3)}$$

3.2 Solving Equation For One Body System

We have the equation of trajectory for one body system. We will use RK-4 method to solve it. We'll convert equation to single derivative form which will be solvable by RK4:-
Let's take $\dot{r} = v$ and $\dot{\phi} = \omega$ so rewriting equation(1)

$$\dot{\omega} = \frac{-2v\omega}{r} \qquad \text{(......eq.4)}$$

and from equation(2)

$$\dot{v} = -\frac{-GM}{r^2} + r\omega^2 \qquad \text{(......eq.5)}$$

We get solution in r and ϕ and then we convert it in cartesian coordinate (x,y) and depending upon initial condition of mass (M),angular velocity(ω), velocity(v), position(r) we get different scenarios. We can also see that equations do not depend on mass (m) of the body in motion.

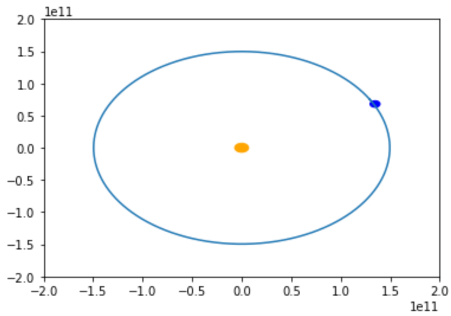
3.3 Factors affecting Dynamics And Their Applications

We know that solution heavily depends on the initial condition of all variables of the equation. So to study effect of mass of stationary body, we take our Earth-Sun system and apply their known initial conditions of angular velocity($\omega = 1.990986 \times 10^7 s^{-1}$), position($r = 1.496 \times 10^{11}$ m), and angle($\phi = \frac{\pi}{6}$).

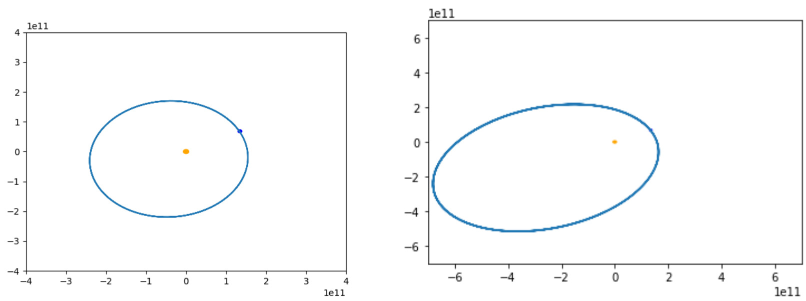
3.3.1 Effect Of Mass Of Stationary Body On Dynamics

Now, we consider moving mass to be earth (fixed mass) and see effect of mass(variable) of stationary body(sun) on dynamics.

- If M= m_{sun} (real mass of sun) ,then earth revolve around revolve sun in circular orbit.

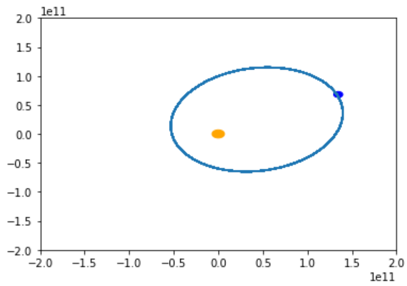


- If M is decrease from m_{sun} then earth revolve in elliptical orbit and with $r_{max} > r_{initial}$,$r_{min} = r_{initial}$ and as decreasing mass r_{max} increases.

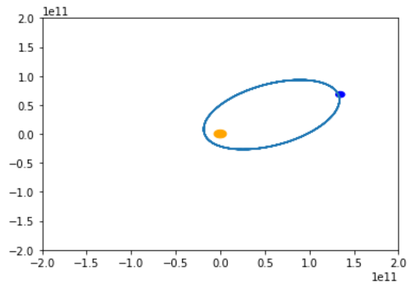


(a) $M_{sun} = 0.8\ m_{sun}$ (b) $M_{sun} = 0.6\ m_{sun}$

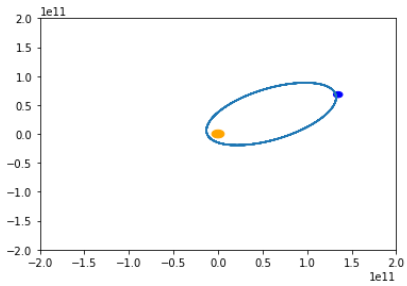
- If M is increased from m_{sun} then also earth revolves in elliptical orbit but with $r_{min} < r_{initial}$ and $r_{max} > r_{initial}$ and as increasing mass r_{min} decrease so from eq.2 $\dot{\phi}$(angular velocity) increases and at further increase earth falls into the sun.



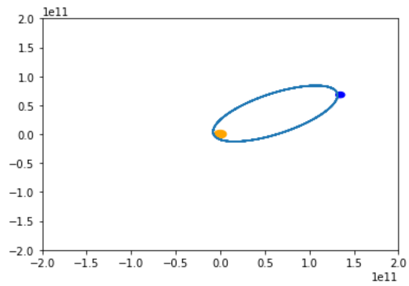
(a) $M_{sun} = 2\ m_{sun}$



(b) $M_{sun} = 5\ m_{sun}$



(c) $M_{sun} = 7\ m_{sun}$

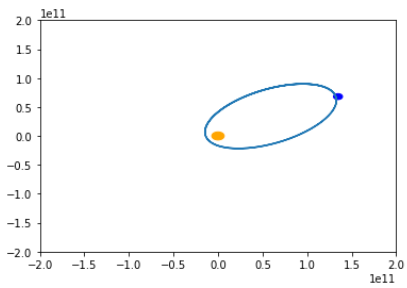


(d) $M_{sun} = 11\ m_{sun}$

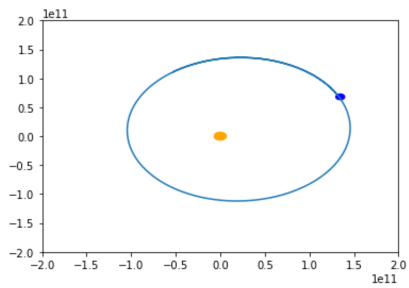
3.3.2 Effect Of Angular Velocity On Dynamics

To see effect of initial angular velocity of earth(moving mass) on dynamics we keep all other variables constant and vary angular velocity.

- If we start from minimum value of initial angular velocity zero(earth would fell into sun) and by increasing the value ,earth starts to orbit around the sun in elliptical path $r_{max} = r_{initial}$ and r_{min} increasing.

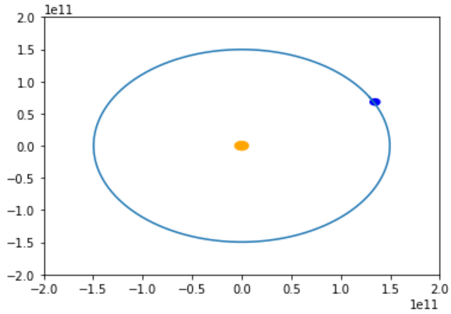


(a) ω=0.4 ω_{earth}



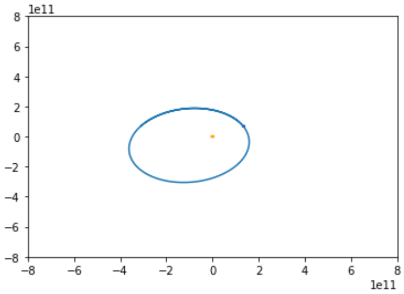
(b) ω=0.9 ω_{earth}

- By increasing it further, the orbit tends to circular orbit from elliptical orbit [from eq]. $\ddot{r} = 0$, $r_{min} = r_{max}$ and E $= U_{eff}min < 0$] and at angular velocity($\omega = 1.990986 \times 10^7\ s^{-1}$) earth orbits in circular path .

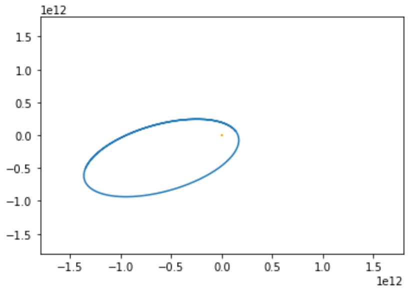


$$\omega = \omega_{earth} = 1.990986 \times 10^{7} s^{-1}$$

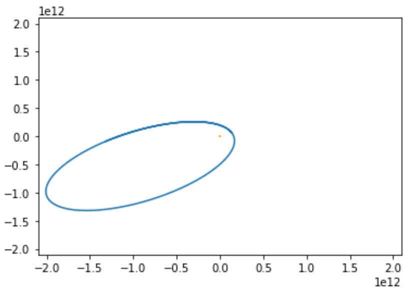
- Further increasing value of angular velocity ω, earth orbits in elliptical path but $r_{min} = r_{initial}$ and r_{max} increase continuously.



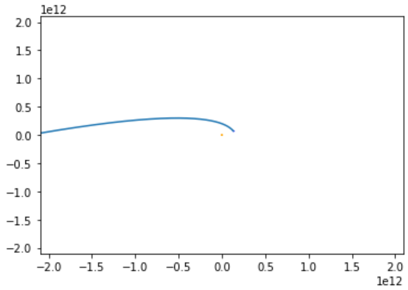
(a) ω=1.2 ω_{earth}



(b) ω=1.35 ω_{earth}



(c) ω= 1.37 ω_{earth}



(d) ω=$\sqrt{2}$ ω_{earth}

- As we can see, orbit of the earth becomes more elliptical by increasing the angular velocity of earth. For one particular angular velocity, earth will escape the gravitational field of sun and its orbit will become unbounded. This technique was also used in **Mangalyaan Project** in which Mangalyaan was set in elliptical orbit and whenever it crossed the perigee, engine was switched on and gave more velocity to mangalyaan which made the elliptical orbit grow and finally it achieved the velocity at perigee equal to escape velocity from earth's gravitational field. In this way we save lot of fuel and solve problem of escaping gravitational field of earth.

4 The Two Body Problem

The two-body problem is to predict the motion of two massive objects which are abstractly viewed as point particles. The problem assumes that the two objects interact only with one another; the only force affecting each object arises from the other one, and all other objects are ignored.

The most prominent case of the classical two-body problem is the gravitational case, arising in astronomy for predicting the orbits (or escapes from orbit) of objects such as satellites, planets, and stars. A two-point-particle model of such a system nearly always describes its behavior well enough to provide useful insights and predictions.

The two-body problem is interesting in astronomy because pairs of astronomical objects are often moving rapidly in arbitrary directions (so their motions become interesting), widely separated from one another (so they won't collide) and even more widely separated from other objects (so outside influences will be small enough to be ignored safely).

Under the force of gravity, each member of a pair of such objects will orbit their mutual center of mass in an elliptical pattern, unless they are moving fast enough to escape one another entirely, in which case their paths will diverge along other planar conic sections. If one object is very much heavier than the other, it will move far less than the other with reference to the shared center of mass. The mutual center of mass may even be inside the larger object.

4.1 Equation Of Motion In Inertial Frame

Figure 1 shows two-point masses acted upon only by the mutual force of gravity between them. The positions R1 and R2 of their centers of mass are shown relative to an inertial frame of reference XYZ. In terms of the coordinates of the two points

$$R_1 = X_1\hat{\imath} + Y_1\hat{\jmath} + Z_1\hat{k} \tag{1}$$

$$R_2 = X_2\hat{\imath} + Y_2\hat{\jmath} + Z_2\hat{k} \tag{2}$$

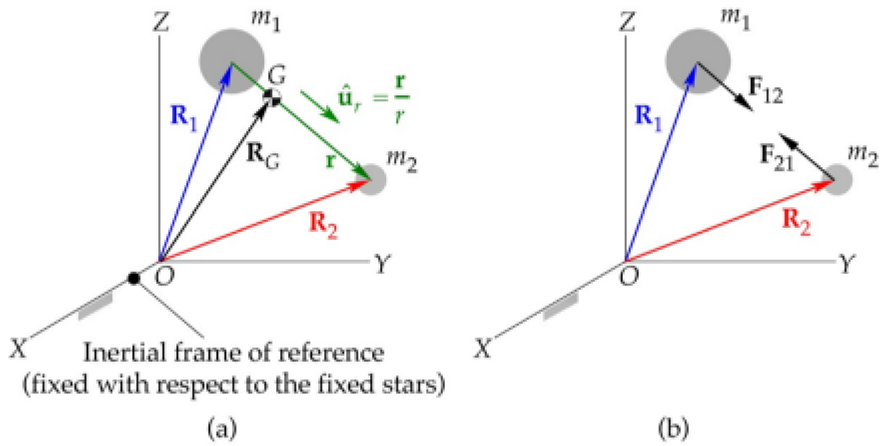


Fig 1 (a) Two masses located in an inertial frame. (b) Free-body diagrams.

The origin O of the inertial frame may move with a constant velocity (relative to the fixed stars), but the axes do not rotate. Each of the two bodies is acted upon by the gravitational attraction of the other. F_{12} is the force exerted on m_1 by m_2, and F_{21} is the force exerted on m_2 by m_1.

The position vector R_G of the center of mass (or barycenter) G of the system in Fig. 1(a) is defined by the formula

$$R_G = \frac{m_1 R_1 + m_2 R_2}{m_1 + m_2} \tag{3}$$

Therefore, the absolute velocity and the absolute acceleration of G are

$$v_G = \dot{R}_G = \frac{m_1 \dot{R}_1 + m_2 \dot{R}_2}{m_1 + m_2} \tag{4}$$

$$a_G = \ddot{R}_G = \frac{m_1 \ddot{R}_1 + m_2 \ddot{R}_2}{m_1 + m_2} \tag{5}$$

Let r be the position vector of m_2 relative to m_1. Then,

$$\mathbf{r} = \mathbf{R_2} - \mathbf{R_1} \tag{6}$$

Furthermore, let $\hat{u}_r$ be the unit vector pointing from m_1 toward m_2, so that

$$\hat{u}_r = \frac{\mathbf{r}}{r} \tag{7}$$

where r is the magnitude of $\mathbf{r}$.

The body m_1 is acted upon only by the force of gravitational attraction toward m_2. The force of gravitational attraction, F_g, which acts along the line joining the centers of mass of m1 and m2, is given by

$$F_{12} = \frac{G m_1 m_2}{r^2} \hat{u}_r \tag{8}$$

where $\hat{u}_r$ accounts for the fact that the force vector F_{12} is directed from m_1 toward m_2. By Newton's third law (the action–reaction principle), the force F_{21} exerted on m_2 by m_1 is $-F_{12}$, so that

$$F_{21} = -\frac{Gm_1m_2}{r^2}\hat{u}_r \tag{9}$$

Newton's second law of motion as applied to a body m_1 is $F_{12} = m_1\ddot{R}_1$, where $\ddot{R}_1$ is the absolute acceleration of m_1. Combining this with Newton's law of gravitation Eq. (8) yields

$$m_1\ddot{R}_1 = \frac{Gm_1m_2}{r^2}\hat{u}_r \tag{10}$$

Likewise,

$$m_2\ddot{R}_2 = -\frac{Gm_1m_2}{r^2}\hat{u}_r \tag{11}$$

It is apparent upon forming the sum of Eqs. (10) and (11) that $m_1\ddot{R}_1 + m_2\ddot{R}_2 = 0$. According to Eq. (5), this means that the acceleration of the center of mass G of the system of two bodies m_1 and m_2 is zero. Therefore, as is true for any system that is free of external forces, G moves in a straight line through space with a constant velocity v_G. Its position vector relative to XYZ is given by

$$R_G = R_{G0} + v_Gt \tag{12}$$

where R_{G0} is the position of G at time $t = 0$. The non accelerating center of mass of a two-body system may serve as the origin of an inertial frame.

The potential energy V of the gravitational force F between two point masses m_1 and m_2 separated by a distance r is given by

$$V = -\frac{Gm_1m_2}{r^2} \tag{13}$$

A conservative force, like gravity, can be obtained from its potential energy function V by means of the gradient operator,

$$F = -\nabla V \tag{14}$$

where, in Cartesian coordinates,

$$\nabla = \frac{\partial}{\partial x}\hat{i} + \frac{\partial}{\partial y}\hat{j} + \frac{\partial}{\partial z}\hat{k} \tag{15}$$

For the two-body system in Fig. 1 we have, by Eqn.(13),

$$V = -\frac{Gm_1m_2}{\sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2 + (Z_2 - Z_1)^2}} \tag{16}$$

The attractive forces F_{12} and F_{21} are derived from Eq. (16) as follows:

$$F_{12} = -\left(\frac{\partial V}{\partial X_2}\hat{i} + \frac{\partial V}{\partial Y_2}\hat{j} + \frac{\partial V}{\partial Z_2}\hat{k}\right) \tag{17}$$

$$F_{21} = -\left(\frac{\partial V}{\partial X_1}\hat{i} + \frac{\partial V}{\partial Y_1}\hat{j} + \frac{\partial V}{\partial Z_1}\hat{k}\right) \tag{18}$$

Let us return to Eqs. (10) and (11), the equations of motion of the two-body system relative to the XYZ inertial frame. We can divide m_1 out of Eq. (10) and m_2 out of Eq. (11) and then substitute Eq. (7) into both results to obtain

$$\ddot{R}_1 = Gm_2\frac{\mathbf{r}}{r^3} \tag{19}$$

$$\ddot{R}_2 = Gm_1\frac{\mathbf{r}}{r^3} \tag{20}$$

These are the final forms of the equations of motion of the two bodies in inertial space. We can express these equations in terms of the components of the position and acceleration vectors in the inertial XYZ frame:

$$\ddot{X}_1 = Gm_2\frac{X_2 - X_1}{r^3} \qquad \ddot{Y}_1 = Gm_2\frac{Y_2 - Y_1}{r^3} \qquad \ddot{Z}_1 = Gm_2\frac{Z_2 - Z_1}{r^3} \tag{21}$$

$$\ddot{X}_2 = Gm_1\frac{X_1 - X_2}{r^3} \qquad \ddot{Y}_2 = Gm_1\frac{Y_1 - Y_2}{r^3} \qquad \ddot{Z}_2 = Gm_1\frac{Z_1 - Z_2}{r^3} \tag{22}$$

The position vector R and velocity vector V of a particle are referred to collectively as its state vector. The fundamental problem before us is to find the state vectors of both particles of the two-body system at a given time given the state vectors at an initial time. The numerical solution procedure is outlined in Algorithm below.

4.2 Algorithm

The state vectors R_1, R_2, V_1, V_2 of the two-body system were numerically calculated as a function of time, given their initial values R_1^0, R_2^0, V_1^0, V_2^0. This algorithm was implemented in Python which is listed in Appendix.

1. Form the vector consisting of the components of the state vectors at time t_0,
$$y_0 = \left[X_1^0, Y_1^0, Z_1^0, X_2^0, Y_2^0, Z_2^0, \dot{X}_1^0, \dot{Y}_1^0, \dot{Z}_1^0, \dot{X}_2^0, \dot{Y}_2^0, \dot{Z}_2^0\right]$$

2. Provide y0 and the final time t_f to Algorithms 1.1, 1.2, or 1.3, along with the vector that comprises the components of the state vector derivatives
$$f(t, y) = \left[\dot{X}_1, \dot{Y}_1, \dot{Z}_1, \dot{X}_2, \dot{Y}_2, \dot{Z}_2, \ddot{X}_1, \ddot{Y}_1, \ddot{Z}_1, \ddot{X}_2, \ddot{Y}_2, \ddot{Z}_2\right]$$
where the last six components, the accelerations, are given by Eqs. (21) and (22).

3. The selected algorithm solves the system $\dot{y} = f(t, y)$ for the system state vector
$$y = \left[X_1, Y_1, Z_1, X_2, Y_2, Z_2, \dot{X}_1, \dot{Y}_1, \dot{Z}_1, \dot{X}_2, \dot{Y}_2, \dot{Z}_2\right]$$
at n discrete times t_n from t_0 through t_f.

4. The state vectors of m_1 and m_2 at the discrete times are

$$R_1 = X_1\hat{i} + Y_1\hat{j} + Z_1\hat{k} \qquad R_2 = X_2\hat{i} + Y_2\hat{j} + Z_2\hat{k}$$

$$V_1 = \dot{X}_1\hat{i} + \dot{Y}_1\hat{j} + \dot{Z}_1\hat{k} \qquad V_2 = \dot{X}_2\hat{i} + \dot{Y}_2\hat{j} + \dot{Z}_2\hat{k}$$

4.3 Output And Analysis

4.3.1 Two Bodies With Same Mass

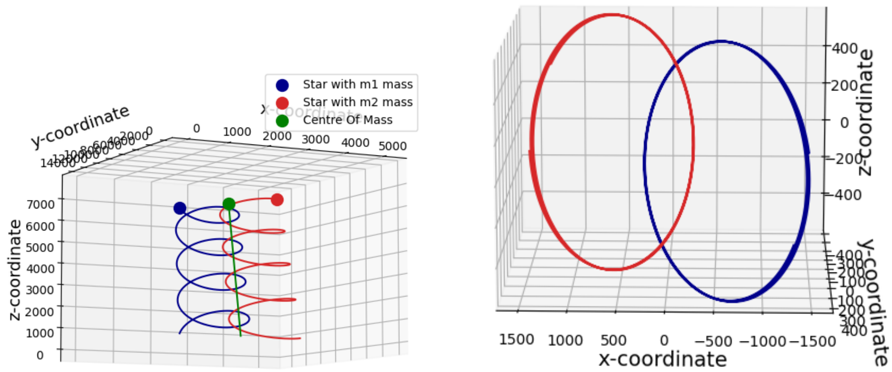


Fig 2 (a) Two masses moving in an inertial frame.(b) Centre of mass frame.

Figure 2 shows the trajectory of two bodies having equal masses. In figure 2(a) it can be in seen that m_1 and m_2 are soon established in a periodic helical motion around the straight-line trajectory of the center of mass G through space. This pattern continues indefinitely. When viewed from the centre of mass frame (Fig 2.b), masses m_1 and m_2 follow elliptical path around the centre of mass. Also since it is assumed that net external force on the system is zero, centre of mass remains stationary.

4.3.2 Binary Star System: ALPHA CENTAURI

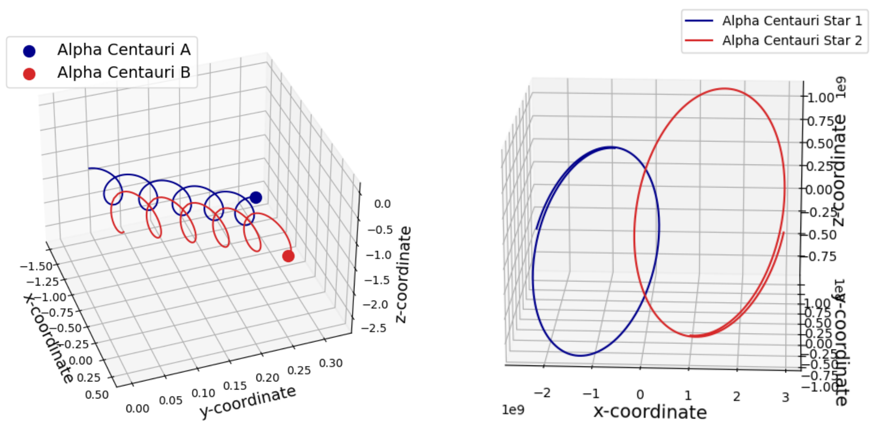
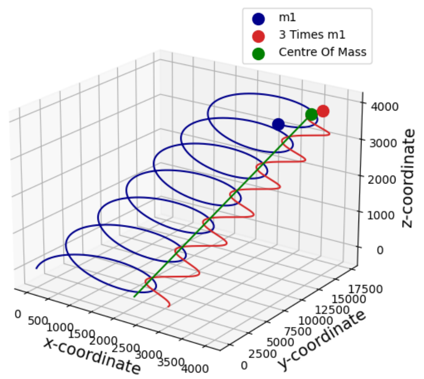


Fig 3 (a) Binary stars moving in an inertial frame.(b) Centre of mass frame.

Figure 3 shows the trajectory of the alpha centauri binary star system. The closest stars to Earth are three stars in the Alpha Centauri system. The two main stars other than the sun are Alpha Centauri A and Alpha Centauri B, which form a binary pair. They are an average of 4.3 light-years from Earth. The third star is Proxima Centauri. It is about 4.22 light-years from Earth.

Alpha centauri A is more massive than alpha centauri B. From their trajectory, we can see that alpha centauri B revolves around the centre of mass at higher speed as compared to alpha centauri A. Also it has a longer trajectory as can be seen by the trajectory depicted in red in figure 3 (b).

4.3.3 Two Objects With Different Masses



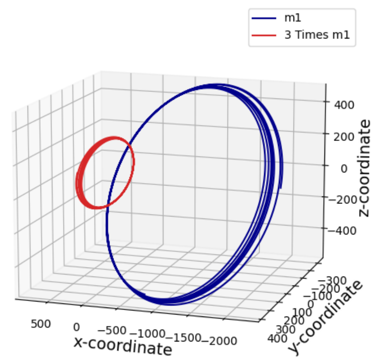


Fig 4 (a) Two objects with different masses.(b) Centre of mass frame.

Figure 4 shows the trajectory of two objects with different masses. The mass of the object in red is three times the blue object. It can be seen that trajectory of blue object is an elongated ellipse and very large as compared to the trajectory of object in red. The red object revolves closer to the centre of mass and at a pace much less than the object in blue. Also, since the net external force on the system is zero, the centre of mass remain stationary (depicted by green in figure 4.a).

5 Three Body Problem

5.1 Equations Of Motion In Inertial Frame

Figure 1 shows three-point masses acted upon only by the mutual force of gravity between them. The positions R1, R2 and R3 of their centers of mass are shown relative to an inertial frame of reference XYZ. In terms of the coordinates of the two points

$$\mathbf{R_1} = X_1\hat{i} + Y_1\hat{j} + Z_1\hat{k} \tag{21}$$

$$\mathbf{R_2} = X_2\hat{i} + Y_2\hat{j} + Z_2\hat{k} \tag{22}$$

$$\mathbf{R_3} = X_3\hat{i} + Y_3\hat{j} + Z_3\hat{k} \tag{23}$$

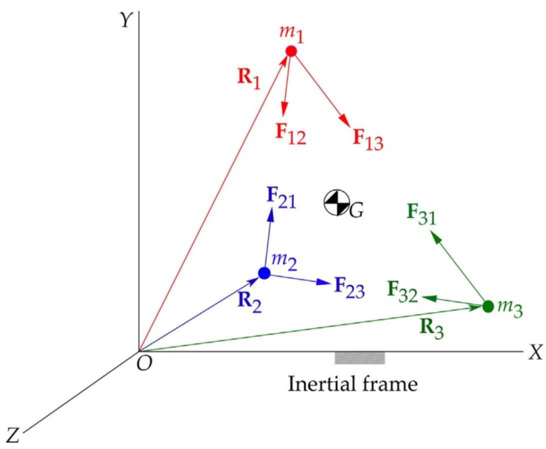


Figure:1 Three body diagram in Inertial Frame

The origin O of the inertial frame may move with a constant velocity (relative to the fixed stars), but the axes do not rotate. Each of the three bodies is acted upon by the gravitational attraction of the other. F_{12} is the force exerted on m_1 by m_2, and F_{21} is the force exerted on m_2 by m_1, similarly force F_{23} and F_{32} for mass m_2 and m_3 and F_{13} and F_{31} for mass m_1 and m_3.

The position vector R_G of the center of mass G of the system in Fig. 1 is defined by the formula

$$\mathbf{R_G} = \frac{m_1\mathbf{R_1} + m_2\mathbf{R_2} + m_3\mathbf{R_3}}{m_1 + m_2 + m_3} \tag{24}$$

Therefore, the absolute velocity and the absolute acceleration of G are

$$\mathbf{v_G} = \dot{\mathbf{R_G}} = \frac{m_1\dot{\mathbf{R_1}} + m_2\dot{\mathbf{R_2}} + m_3\dot{\mathbf{R_3}}}{m_1 + m_2 + m_3} \tag{25}$$

$$\mathbf{a_G} = \ddot{\mathbf{R_G}} = \frac{m_1\ddot{\mathbf{R_1}} + m_2\ddot{\mathbf{R_2}} + m_3\ddot{\mathbf{R_3}}}{m_1 + m_2 + m_3} \tag{26}$$

Let r_1 is the position vectorof m_1 relative to m_2

$$\mathbf{r_1} = \mathbf{R_2} - \mathbf{R_1} \tag{27}$$

Let r_2 is the position vectorof m_2 relative to m_3

$$\mathbf{r_2} = \mathbf{R_3} - \mathbf{R_2} \tag{28}$$

Let r_3 is the position vectorof m_1 relative to m_3

$$\mathbf{r_3} = \mathbf{R_1} - \mathbf{R_3} \tag{29}$$

Furthermore, let $\hat{u}_r$ be the unit vector pointing in the direction of r_i so that

$$\hat{u}_{r_i} = \frac{\mathbf{r_i}}{r_i} \tag{30}$$

where r_i is the magnitude of $\mathbf{r_i}$.
The only force between mass m_1, m_2 and m_3 is gravitational force and this is given by,

$$\mathbf{F_{12}} = \frac{Gm_1m_2}{r_1^2}\hat{u}_{r_1} \tag{31}$$

$$\mathbf{F_{23}} = \frac{Gm_2m_3}{r_2^2}\hat{u}_{r_2} \tag{32}$$

$$\mathbf{F_{13}} = \frac{Gm_1m_3}{r_3^2}\hat{u}_{r_3} \tag{33}$$

where $\hat{u}_{r_i}$ accounts for the fact that the force vector F_{ij} is directed from m_i toward m_j. By Newton's third law (the action–reaction principle), the force F_{ij} exerted on m_i by m_j is $-F_{ji}$. The equation of motion of body 1 is,

$$\mathbf{F_{12}} + \mathbf{F_{13}} = m_1\mathbf{a_1} \tag{34}$$

Substituting Eqs. (11) and (13) yields,

$$\mathbf{a_1} = \frac{Gm_2}{r_1^2}\hat{u}_{r_1} + \frac{Gm_3}{r_3^2}\hat{u}_{r_3}$$

For bodies 2 and 3 we find in a similar fashion that,

$$\mathbf{a_2} = -\frac{Gm_1}{r_1^2}\hat{u}_{r_1} + \frac{Gm_3}{r_2^2}\hat{u}_{r_2}$$

$$\mathbf{a_3} = \frac{Gm_1}{r_3^2}\hat{u}_{r_3} - \frac{Gm_2}{r_2^2}\hat{u}_{r_2}$$

The velocities are related to the accelerations by,

$$\frac{d\mathbf{v_i}}{dt} = \mathbf{a_i}$$

and the position vectors are likewise related to the velocities,

$$\frac{d\mathbf{R_i}}{dt} = \mathbf{v_i}$$

where i = 1,2,3
Eqs. $\mathbf{a_i}$ constitute a system of ordinary differential equations (ODEs) in variable time,
We first resolve all the vectors into their three components along the XYZ axes of the inertial frame and write them as column vectors

$$\mathbf{R_1} = \begin{pmatrix} x_1 & y_1 & z_1 \end{pmatrix}^T$$
$$\mathbf{R_2} = \begin{pmatrix} x_2 & y_2 & z_2 \end{pmatrix}^T$$
$$\mathbf{R_3} = \begin{pmatrix} x_3 & y_3 & z_3 \end{pmatrix}^T$$

$$\mathbf{v_1} = \begin{pmatrix} \dot{x_1} & \dot{y_1} & \dot{z_1} \end{pmatrix}^T$$
$$\mathbf{v_2} = \begin{pmatrix} \dot{x_2} & \dot{y_2} & \dot{z_2} \end{pmatrix}^T$$
$$\mathbf{v_3} = \begin{pmatrix} \dot{x_3} & \dot{y_3} & \dot{z_3} \end{pmatrix}^T$$

According to Eqs. of $\mathbf{a_i}$s,

$$\mathbf{a_1} = \begin{pmatrix} \ddot{x_1} \\ \ddot{y_1} \\ \ddot{z_1} \end{pmatrix} = \begin{pmatrix} \frac{Gm_2(x_2 - x_1)}{r_1^3} + \frac{Gm_3(x_3 - x_1)}{r_3^3} \\ \frac{Gm_2(y_2 - y_1)}{r_1^3} + \frac{Gm_3(y_3 - y_1)}{r_3^3} \\ \frac{Gm_2(z_2 - z_1)}{r_1^3} + \frac{Gm_3(z_3 - z_1)}{r_3^3} \end{pmatrix}$$

$$\mathbf{a_2} = \begin{pmatrix} \ddot{x_2} \\ \ddot{y_2} \\ \ddot{z_2} \end{pmatrix} = \begin{pmatrix} \frac{Gm_1(x_1 - x_2)}{r_1^3} + \frac{Gm_3(x_3 - x_2)}{r_2^3} \\ \frac{Gm_1(y_1 - y_2)}{r_1^3} + \frac{Gm_3(y_3 - y_2)}{r_2^3} \\ \frac{Gm_1(z_1 - z_2)}{r_1^3} + \frac{Gm_3(z_3 - z_2)}{r_2^3} \end{pmatrix}$$

$$\mathbf{a_3} = \begin{pmatrix} \ddot{x_3} \\ \ddot{y_3} \\ \ddot{z_3} \end{pmatrix} = \begin{pmatrix} \frac{Gm_1(x_1 - x_3)}{r_3^3} + \frac{Gm_2(x_2 - x_3)}{r_2^3} \\ \frac{Gm_1(y_1 - y_3)}{r_3^3} + \frac{Gm_2(y_2 - y_3)}{r_2^3} \\ \frac{Gm_1(z_1 - z_2)}{r_3^3} + \frac{Gm_2(z_2 - z_3)}{r_2^3} \end{pmatrix}$$

Next, we form the 18-component column vector

$$\mathbf{y} = \begin{pmatrix} \mathbf{R_1} & \mathbf{R_2} & \mathbf{R_3} & \mathbf{v_1} & \mathbf{v_2} & \mathbf{v_3} \end{pmatrix}^T$$

The first derivatives of the components of this vector comprise the column vector,

$$\dot{\mathbf{y}} = \begin{pmatrix} \mathbf{v_1} & \mathbf{v_2} & \mathbf{v_3} & \mathbf{a_1} & \mathbf{a_2} & \mathbf{a_3} \end{pmatrix}^T$$

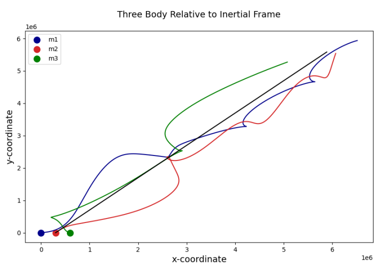
above equation is differntial equation, we can solve it by RK-4 method

$$\dot{\mathbf{y}} = f(t, y)$$

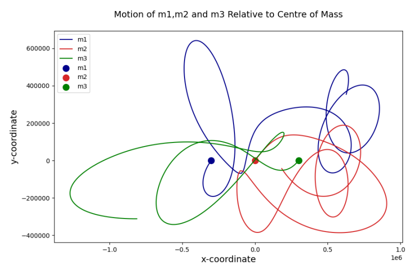
For simplicity, we will solve the three-body problem in the plane. That is, we will restrict ourselves to only the XY components of the vectors **R**, **v**, and **a**.

5.2 Analysis

5.2.1 Three Bodies With Same Mass



(a) Inertial Frame



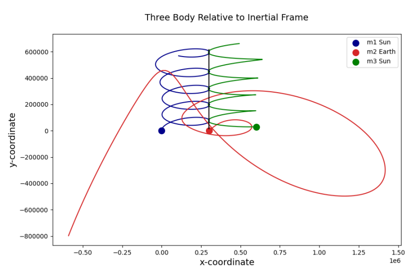
(b) Centre of mass frame

Figure:2 Same Mass

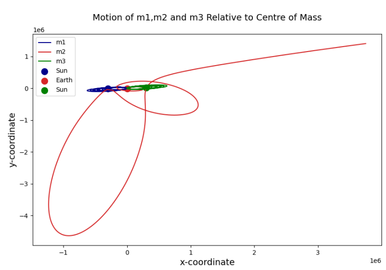
Initial Conditions: m1=1.e29, m2=1.e29, m3=1.e29, t=67000, r1=[0,0], r2=[300000,0], r3=[2*300000,0], v1=[0,0], v2=[250,250], v3=[0,0].

Fig.2 shows the results for three body of equal mass, equally spaced initially along the X axis of an inertial frame. The central mass has an initial velocity in the XY plane, while the other two are at rest. As time progresses, we see no periodic behavior. The chaos is more obvious if the motion is viewed from the center of mass of the three-body system, as shown in Fig.2(b).

5.2.2 Two Sun One Earth



(a) Inertial Frame



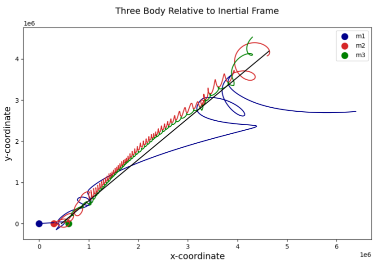
(b) Centre of mass frame

Figure:3 Sun-Earth-Sun

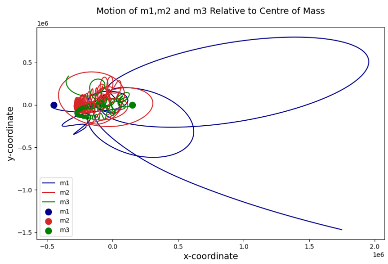
Initial Condetions:m1=1.989e30 m2=5.972e24, m3=1.989e30, au=1.496e8, t=3 years, r1=[0,0], r2=[au,0], r3=[2*au,50000], v1=[0,0], v2=[250,250], v3=[30,0].

Fig.3 shows the results for three body two of them are of same mass and one has less mass. From fig.3(a) one can see that two equal mass sun are orbiting around their centre of mass, while lower mass body orbiting chaotically, that one can see from fig.3(b).From fig.3 one can see that, lower mass body experience gravitational slinger shot three time. After long time it become unbound from the three body system.

5.2.3 Three different Mass



(a) Inertial Frame



(b) Centre of mass frame

Figure:4 Three different mass

Fig4 shows the results for three body with different mass. Fig.4(a) shows that two heavy body orbit with respect to their centre of mass, while lower mass body orbit chaotically. One can see that heaviest body orbits in smaller circle, while other body orbits with higher radius circle. Again after long time body with smallest mass will become unbound as one can see from fig.4

6 Appendix

Here, All codes are in python programming language.
For All Codes Click on Following Link:https://github.com/JemishNaliyapara007/Trajectories-In-The-Universe.git

6.1 Code For One Body Problem:

```
import numpy as np
import matplotlib.pyplot as plt
def f1(r,v,theta,A_v,t):
    return v
def f2(r,v,theta,A_v,t):
    G=6.67408e-11
    M_s=1.989e30
    return r*A_v**2-G*M_s/r**2
def f3(r,v,theta,A_v,t):
    return A_v
def f4(r,v,theta,A_v,t):
    return -2*v*A_v/r

def RK4(f1,f2,f3,f4,r,v,theta,A_v,ti,tf,n):
    dt=(tf-ti)/n
    a=np.zeros(shape=(n+1))
    b=np.zeros(shape=(n+1))
    c=np.zeros(shape=(n+1))
    d=np.zeros(shape=(n+1))
    a[0]=r
    b[0]=v
    c[0]=theta
    d[0]=A_v
    t=np.linspace(ti,tf,n+1)
    for i in range(n):
        k11=dt*f1(r,v,theta,A_v,t)
        k21=dt*f2(r,v,theta,A_v,t)
        k31 = dt * f3(r, v, theta, A_v, t)
        k41 = dt * f4(r, v, theta, A_v, t)
        k12 = dt * f1(r+0.5*k11, v+0.5*k21, theta+0.5*k31, A_v+0.5*k41, t+0.5*dt)
        k22 = dt * f2(r+0.5*k11, v+0.5*k21, theta+0.5*k31, A_v+0.5*k41, t+0.5*dt)
        k32 = dt * f3(r + 0.5 * k11, v + 0.5 * k21, theta + 0.5 * k31, A_v + 0.5 * k41, t + 0.5 * dt)
        k42 = dt * f4(r + 0.5 * k11, v + 0.5 * k21, theta + 0.5 * k31, A_v + 0.5 * k41, t + 0.5 * dt)
        k13 = dt * f1(r + 0.5 * k12, v + 0.5 * k22, theta + 0.5 * k32, A_v + 0.5 * k42, t + 0.5 * dt)
        k23 = dt * f2(r + 0.5 * k12, v + 0.5 * k22, theta + 0.5 * k32, A_v + 0.5 * k42, t + 0.5 * dt)
        k33 = dt * f3(r + 0.5 * k12, v + 0.5 * k22, theta + 0.5 * k32, A_v + 0.5 * k42, t + 0.5 * dt)
        k43 = dt * f4(r + 0.5 * k12, v + 0.5 * k22, theta + 0.5 * k32, A_v + 0.5 * k42, t + 0.5 * dt)
        k14 = dt * f1(r + k13, v + k23, theta + k33, A_v + k43, t + dt)
        k24 = dt * f2(r + k13, v + k23, theta + k33, A_v + k43, t + dt)
        k34 = dt * f3(r + k13, v + k23, theta + k33, A_v + k43, t + dt)
        k44 = dt * f4(r + k13, v + k23, theta + k33, A_v + k43, t + dt)
        dr=(k11+2*k12+2*k13+k14)/6
        dv = (k21 + 2 * k22 + 2 * k23 + k24) / 6
```

```
        dtheta = (k31 + 2 * k32 + 2 * k33 + k34) / 6
        dA_v = (k41 + 2 * k42 + 2 * k43 + k44) / 6
        r=r+dr
        v=v+dv
        theta=theta+dtheta
        A_v=A_v+dA_v
        a[i+1] = r
        b[i+1] = v
        c[i+1] = theta
        d[i+1] = A_v
    return t,a,b,c,d


r0=1.496e11
v0=0.00
theta0=np.pi/6
A_v0=1.990986e-7

t,a,b,c,d = RK4(f1,f2,f3,f4,r0,v0,theta0,A_v0,0,3.154e7,10000)
n=10000
x=np.zeros(shape=(n+1))
y=np.zeros(shape=(n+1))
for i in range(n+1):
    x[i]=a[i]*np.cos(c[i])
    y[i]=a[i]*np.sin(c[i])

circle=plt.Circle((0,0),696340e4,color='orange',fill=True)
circle1=plt.Circle((1.347e11,6.80e10),4*12742e5,color='b',fill=True)
fig, ax=plt.subplots()
ax.add_artist(circle)
ax.add_artist(circle1)
plt.xlim(-2e11,2e11)
plt.ylim(-2e11,2e11)
ax.plot(x,y)
plt.show()
```

6.2 Code For Two Body Problem:

```
import numpy as np
import matplotlib.pyplot as plt
import scipy.integrate

def RK4(f1,ti,tf,y0,n):
    t=np.zeros(n)
    h = (tf - ti) / (n - 1)
    t[0] = ti
    y=np.zeros(shape=(len(y0),n))
    for i in range(len(y0)):
        y[i][0]=y0[i]


    for i in range(n-1):
        a=np.zeros(len(y0))
        b = np.zeros(len(y0))
        c = np.zeros(len(y0))
        e = np.zeros(len(y0))
        for k in range(len(y0)):
            e[k] = y[k][i]
        k1=f1(t[i],e)
        for k in range(len(y0)):
            a[k]=y[k][i]+k1[k]*h/2
        k2=h*f1(t[i] + h/2,a)
        for k in range(len(y0)):
            b[k]=y[k][i]+k2[k]*h / 2
        k3=h*f1(t[i]+h/2,b)
        for k in range(len(y0)):
            c[k]=y[k][i] + k3[k] * h
        k4=h*f1(t[i]+h,c)
        for j in range(len(y0)):
            y[j][i+1]=y[j][i]+((k1[j]+2*(k2[j]+k3[j])+k4[j])*h)/6
        t[i+1]=t[i]+h
    return t,y



def f1(t,y):
    R1=np.array([y[0],y[1],y[2]])
    R2 = np.array([y[3], y[4], y[5]])
    V1 = np.array([y[6], y[7], y[8]])
    V2 = np.array([y[9], y[10], y[11]])
    r=np.linalg.norm(R2-R1)
    A1=G*m2*(R2-R1)/r**3
    A2=G*m1*(R1-R2)/r**3
    dydt=np.array([V1,V2,A1,A2]).reshape(12,)
```

```
    return dydt


G=6.67259e-20
m1=1.e26
m2=1.e26
t0=0
tf=480
au=1.496e8
r1=np.array([0.,0.,0.])
r2=np.array([3000,0,0])
v1=np.array([10,20,30]) #230
v2=np.array([0,40,0])


b0=np.array([r1,r2,v1,v2])
y0=b0.reshape(12,)

'''
# For Better Accuracy Use This Block
#t1=np.linspace(t0,tf,2000)
#sol=scipy.integrate.solve_ivp(f1,[t0,tf],y0,method='RK45',t_eval=t1)
#t=sol.t
#y=sol.y
'''


t,y=RK4(f1,t0,tf,y0,1000)
n=len(t)
x=[]
y1=[]
z=[]
for i in range(n):
    x.append((m1*y[0][i]+m2*y[3][i])/(m1+m2))
    y1.append((m1 * y[1][i] + m2 * y[4][i]) / (m1 + m2))
    z.append((m1 * y[2][i] + m2 * y[5][i]) / (m1 + m2))
xg=np.array(x)
yg=np.array(y1)
zg=np.array(z)

fig=plt.figure(figsize=(10,10))

#Here Program Make Three Type Of Graph ,So Remove Comment Block One At Time To Get Plot

ax=fig.add_subplot(1,1,1,projection="3d")
ax.plot(y[0],y[1],y[2],color="darkblue")
ax.plot(y[3],y[4],y[5],color="tab:red")
ax.plot(xg,yg,zg,color="green")
```

```
ax.set_title("Motion Relative to Inertial Frame\n",fontsize=14)
ax.scatter(y[0,-1],y[1,-1],y[2,-1],color="darkblue",marker="o",s=100,label="m1")
ax.scatter(y[3,-1],y[4,-1],y[5,-1],color="tab:red",marker="o",s=100,label="3 Times m1")
ax.scatter(xg[-1],yg[-1],zg[-1],color="green",marker="o",s=100,label="Centre Of Mass")


'''
ax=fig.add_subplot(1,1,1,projection="3d")
ax.plot(y[0]-xg,y[1]-yg,y[2]-zg,color="darkblue",label="m1")
ax.plot(y[3]-xg,y[4]-yg,y[5]-zg,color="tab:red",label="3 Times m1")
ax.set_title("Motion of m1 and m2 Relative to Centre of Mass\n",fontsize=14)
'''
'''
ax=fig.add_subplot(1,1,1,projection="3d")
ax.plot(y[3]-y[0],y[4]-y[1],y[5]-y[2],color="darkblue",label="m2")
ax.plot(xg-y[0],yg-y[1],zg-y[2],color="tab:red",label="Centre of Mass")
ax.set_title("Motion of m2 and Centre of mass relative to m1\n",fontsize=14)
'''

ax.set_xlabel("x-coordinate",fontsize=14)
ax.set_ylabel("y-coordinate",fontsize=14)
ax.set_zlabel("z-coordinate",fontsize=14)
ax.legend()
plt.show()
```

6.3 Code For Three Body Problem:

```
import numpy as np
import matplotlib.pyplot as plt


def RK4(f1,ti,tf,y0,n):
    t=np.zeros(n)
    h = (tf - ti) / (n - 1)
    t[0] = ti
    y=np.zeros(shape=(len(y0),n))
    for i in range(len(y0)):
        y[i][0]=y0[i]


    for i in range(n-1):
        a=np.zeros(len(y0))
        b = np.zeros(len(y0))
        c = np.zeros(len(y0))
        e = np.zeros(len(y0))
        for k in range(len(y0)):
            e[k] = y[k][i]
        k1=f1(t[i],e)
        for k in range(len(y0)):
            a[k]=y[k][i]+k1[k]*h/2
        k2=h*f1(t[i] + h/2,a)
        for k in range(len(y0)):
            b[k]=y[k][i]+k2[k]*h / 2
        k3=h*f1(t[i]+h/2,b)
        for k in range(len(y0)):
            c[k]=y[k][i] + k3[k] * h
        k4=h*f1(t[i]+h,c)
        for j in range(len(y0)):
            y[j][i+1]=y[j][i]+((k1[j]+2*(k2[j]+k3[j])+k4[j])*h)/6
        t[i+1]=t[i]+h
    return t,y

def f1(t,y):
    R1=np.array([y[0],y[1]])
    R2 = np.array([y[2], y[3]])
    R3=np.array([y[4],y[5]])
    V1 = np.array([y[6], y[7]])
    V2 = np.array([y[8], y[9]])
    V3 = np.array([y[10], y[11]])
    r12=np.linalg.norm(R2-R1)
    r13 = np.linalg.norm(R3 - R1)
    r23 = np.linalg.norm(R2 - R3)
```

```
    A1=G*m2*(R2-R1)/r12**3 + G*m3*(R3-R1)/r13**3
    A2=G*m1*(R1-R2)/r12**3 + G*m3*(R3-R2)/r23**3
    A3 = G * m1 * (R1 - R3) / r13 ** 3 + G * m2 * (R2 - R3) / r23 ** 3
    dydt=np.array([V1,V2,V3,A1,A2,A3]).reshape(12,)
    return dydt


G=6.67259e-20
m1=1.e29
m2=1.e29
m3=1.e29
t0=0
tf=67000
au=1.4932e8
r1=np.array([0,0])
r2=np.array([300000,0])
r3=np.array([2*300000,0])
v1=np.array([0,0])
v2=np.array([250,250])
v3=np.array([0,0])

m=m1+m2+m3
b0=np.array([r1,r2,r3,v1,v2,v3])
y0=b0.reshape(12,)

t,y=RK4(f1,t0,tf,y0,100000)
n=len(t)

x=[]
y1=[]
for i in range(n):
    x.append((m1*y[0][i]+m2*y[2][i]+m3*y[4][i])/m)
    y1.append((m1 * y[1][i] + m2 * y[3][i]+m3*y[5][i])/m)
xg=np.array(x)
yg=np.array(y1)

fig=plt.figure(figsize=(10,10))
#Here Program Make Three Type Of Graph ,So Remove Comment Block One At Time To Get Plot

ax=fig.add_subplot(1,1,1)
ax.plot(y[0],y[1],color="darkblue")
ax.plot(y[2],y[3],color="tab:red")
ax.plot(y[4],y[5],color="green")
ax.plot(xg,yg,color="k")
ax.set_title("Three Body Relative to Inertial Frame\n",fontsize=14)
ax.scatter(y[0,0],y[1,0],color="darkblue",marker="o",s=100,label="m1")
ax.scatter(y[2,0],y[3,0],color="tab:red",marker="o",s=100,label="m2 ")
ax.scatter(y[4,0],y[5,0],color="green",marker="o",s=100,label="m3 ")
```

```
'''
ax=fig.add_subplot(1,1,1)
ax.plot(y[0]-xg,y[1]-yg,color="darkblue",label="m1")
ax.plot(y[2]-xg,y[3]-yg,color="tab:red",label="m2")
ax.plot(y[4]-xg,y[5]-yg,color="green",label="m3")
ax.set_title("Motion of m1,m2 and m3 Relative to Centre of Mass\n",fontsize=14)
ax.scatter(y[0,0]-xg[0],y[1,0]-yg[0],color="darkblue",marker="o",s=100,label="m1")
ax.scatter(y[2,0]-xg[0],y[3,0]-yg[0],color="tab:red",marker="o",s=100,label="m2")
ax.scatter(y[4,0]-xg[0],y[5,0]-yg[0],color="green",marker="o",s=100,label="m3")
'''

'''
ax=fig.add_subplot(1,1,1)
ax.plot(y[2]-y[0],y[3]-y[1],color="darkblue",label="m2")
ax.plot(y[4]-y[0],y[5]-y[1],color="green",label="m3")
ax.plot(xg-y[0],yg-y[1],color="k",label="Centre of Mass")
ax.set_title("Motion of m2 ,m3 and Centre of mass relative to m1\n",fontsize=14)
'''
ax.set_xlabel("x-coordinate",fontsize=14)
ax.set_ylabel("y-coordinate",fontsize=14)
ax.legend()
plt.show()
```

6.4 Runge-Kutta Method (RK4):

Runge-Kutta Method(RK4) is used to solve differential equations. In this code, we use array of differential equations (as you seen in Appendix 2 and Appendix 3). This code treat every element of array as individual function and solve it using RK4 method.

```
import numpy as np


def RK4(f1,ti,tf,y0,n):
    t=np.zeros(n)
    h = (tf - ti) / (n - 1)
    t[0] = ti
    y=np.zeros(shape=(len(y0),n))
    for i in range(len(y0)):
        y[i][0]=y0[i]


    for i in range(n-1):
        a=np.zeros(len(y0))
        b = np.zeros(len(y0))
        c = np.zeros(len(y0))
        e = np.zeros(len(y0))
        for k in range(len(y0)):
            e[k] = y[k][i]
        k1=f1(t[i],e)
        for k in range(len(y0)):
            a[k]=y[k][i]+k1[k]*h/2
        k2=h*f1(t[i] + h/2,a)
        for k in range(len(y0)):
            b[k]=y[k][i]+k2[k]*h / 2
        k3=h*f1(t[i]+h/2,b)
        for k in range(len(y0)):
            c[k]=y[k][i] + k3[k] * h
        k4=h*f1(t[i]+h,c)
        for j in range(len(y0)):
            y[j][i+1]=y[j][i]+((k1[j]+2*(k2[j]+k3[j])+k4[j])*h)/6
        t[i+1]=t[i]+h
    return t,y
```

6.5 Simulations

1.Sun Earth System : https://youtu.be/Ta1sS3A14-c
2.Sun Earth System [M=$0.8 \times M_{sun}$] : https://youtu.be/riyFb_ezUMg
3.Sun Earth System [M=$6 \times M_{sun}$] : https://youtu.be/qaUgtXyX4kw
4.Two Bodies System [Alpha Centauri System] : https://youtu.be/R2FHCoKgk3U
5.Two Bodies System [Different Mass] : https://youtu.be/yIcj27zrPqo
6.Three Bodies System [Same Mass] : https://youtu.be/kmmhFSy5VSo
7.Three Bodies System [Different Mass] : https://youtu.be/qHkyoFLYZWg

7 Bibliography

References

[1] Howard D.Curtis *Orbital Mechanics for Engineering Student*. Butterworth-Heinemann, Elsevier, 2020.

[2] John R. Taylor *Classical Mechanics*. University Science Books,ISBN 1-891389-22-X, 2005

[3] Website:Programming a simulation of the Earth orbiting the Sun, https://evgenii.com/blog/earth-orbit-simulation/

[4] Website:Modelling the Three Body Problem in Classical Mechanics using Python, https://towardsdatascience.com/modelling-the-three-body-problem-in-classical-mechanics-