

Black Hole Orbital Trajectory Final Report

Ian Taylor, Michelle Li, Mickey Shamah,
Gerardo Elguezabal, Yuliya Gumenyuk

May 15, 2024

1 Project Overview

The project aims to explore black hole orbits, focusing on understanding how angular momentum and energy influences the motion of objects around it. Through theoretical analysis and numerical simulations, the project delves into topics such as the effective potential and various types of orbits that can exist in the vicinity of a black hole. As well as the calculation of the trajectory of an object on a given orbit.

2 Theoretical Background

2.1 Understanding Schwarzschild Orbits and Effective Potential

For this project we will consider Schwarzschild orbits because they can be described as one-dimensional motion in an effective potential. The Schwarzschild solution describes the spacetime around a spherically symmetric, non-rotating mass, which is often used to model black holes. In these calculations, G represents the gravitational constant and M represents the mass of the black hole. Setting these constants to 1 simplifies the equations significantly.

The effective potential for a Schwarzschild black hole is derived from the geodesic equations of motion for test particles. These equations describe how particles move along paths in curved spacetime. By rearranging terms in these equations, we can define an effective potential that captures the effects of gravity on the motion of the particle.

For the Schwarzschild metric, the effective potential V_{eff} for a particle with $M = 1$ and angular momentum L moving around the black hole is given by:

$$V_{\text{eff}}(r) = \frac{1}{2} - \frac{1}{r} + \frac{L^2}{2r^2} - \frac{L^2}{r^3}$$

Here r is the radial coordinate of the particle's orbit.

The effective potential helps in understanding the behavior of test particles around the black hole.

2.2 Effective Potential for different L values

We can divide the effective potential graphs into 5 different cases.

1. $L < L_{\text{ISCO}}$ (Innermost Stable Circular Orbit):

- In this case, the particle's angular momentum is less than the critical value L_{ISCO} required for stable circular orbits at the Innermost Stable Circular Orbit.
- The effective potential $V_{\text{eff}}(r)$ will have a single minimum. Particles with this angular momentum will have stable bound orbits around the black hole at radii larger than r_{ISCO} .
- These particles will oscillate around the minimum of the potential, representing stable, periodic motion.

2. $L = L_{\text{ISCO}}$ (Innermost Stable Circular Orbit):

- When $L = L_{\text{ISCO}}$, the effective potential will have a single minimum, just like in the previous case.
- Particles with this specific angular momentum will be able to maintain a stable circular orbit at the Innermost Stable Circular Orbit.

3. $L_{ISCO} < L < L_{IBCO}$ (**Between Innermost Stable Circular Orbit and Innermost Bound Circular Orbit**):

- In this range, the effective potential will have two minima. One minimum will correspond to stable circular orbits, while the other will correspond to unstable circular orbits closer to the black hole.
- Particles with angular momentum in this range can occupy either stable or unstable circular orbits, depending on their initial conditions.

4. $L = L_{IBCO}$ (**Innermost Bound Circular Orbit**):

- When $L = L_{IBCO}$, the effective potential will have two identical minima.
- Particles with this specific angular momentum will be able to maintain a circular orbit at the Innermost Bound Circular Orbit, which is the smallest stable circular orbit possible around the black hole.

5. $L > L_{IBCO}$ (**Beyond Innermost Bound Circular Orbit**):

- For angular momenta greater than the critical value L_{IBCO} , the effective potential will only have one minimum, corresponding to a stable circular orbit at a radius larger than the Innermost Bound Circular Orbit.
- Particles with angular momentum greater than L_{IBCO} will have stable circular orbits at radii larger than the Innermost Bound Circular Orbit.

These different scenarios illustrate how the angular momentum of a particle affects its motion around a black hole and how it determines the nature of its orbit.

2.3 Understanding Different Types of Orbits Around a Black Hole

In the study of black hole dynamics, based on the energy (E) and angular momentum (L) different types of orbits can arise. Here, we categorize orbits based on these values. We will refer back to the 5 cases of different L values discussed above. First let's establish that E_u stands for value of E that represents an unstable circular orbit, E_s stands for value of E that represents a stable circular orbit, and E_{ISCO} represents the value of E where $E_u = E_s$, then there is only one circular orbit.

2.3.1 Type 1 Orbit (Reaching Both $r = 0$ and $r = \infty$)

Type 1 orbits live above the V_{eff} graph. These orbits are possible for $E \geq 1$ for values of L in cases 1-3, for $E > 1$ for case 4, and for $E > E_u$ for case 5. These type of orbits reach both $r = 0$ and extend outward to $r = \infty$, represent trajectories where test particles initially approach the black hole, reaching a minimum distance before escaping to infinity.

2.3.2 Type 2 Orbits (Reaching $r = \infty$ and have an r_{min})

Type 2 orbits are characterized by trajectories that extend to $r = \infty$. Type 2 orbits are possible for case 5 L values, where $1 \leq E \leq E_u$. Type 2 orbits demonstrate trajectories where particles possess sufficient energy to escape the black hole's gravitational pull entirely.

2.3.3 Type 3 Orbits (Reaching $r = 0$ and has r_{max})

Type 3 orbits are trajectories where test particles approach the black hole and reach $r = 0$. Type 3 orbits are possible for all different cases of L values. For case 1 and 2, type 3 orbits exist if $E < 1$, for case 3 we need $E \leq E_u$ or $E_u < E < 1$, there is a discontinuity at E_u , for case 4 we have $E \leq 1$, and for case 5 type 3 orbits are possible for $E \leq E_u$, since E_u is above $E = 1$.

2.3.4 Type 4 Orbits (with r_{min} and r_{max})

Type 4 orbits are characterized by their finite radial extent, bounded by two distinct radii: r_{min} and r_{max} . These orbits are confined within a certain range of distances from the black hole, determined by the specific values of the particle's energy and angular momentum. Type 4 orbits exist for cases 2-5. For case 2, we need $E = E_{ISCO} = E_u = E_s$, this is the only radii bounded and circular orbit. For case 3 and 4 we need $E_s \leq E \leq E_u$, and for case 5 we want $E_s \leq E < 1$ or $E = E_u$.

2.4 Integration: Applying the SciPy Integrator to a System of ODEs

In this section, we will discuss how to apply numerical integration using the SciPy integrator to solve a system of ordinary differential equations (ODEs) derived from the metric solution for a non-spinning, non-charged black hole.

2.4.1 Defining the Metric Solution and Equations of Motion

For a Schwarzschild black hole, the spacetime is described by the metric:

$$ds^2 = - \left(1 - \frac{2M}{r}\right) dt^2 + \left(1 - \frac{2M}{r}\right)^{-1} dr^2 + r^2(d\theta^2 + \sin^2 \theta d\phi^2),$$

where M is the mass of the black hole.

In the equatorial plane ($\theta = \frac{\pi}{2}$), the equations of motion for a test particle orbiting the black hole can be derived using the geodesic equations. The resulting system of ODEs can be expressed in terms of the proper time τ :

$$\begin{aligned}\frac{dt}{d\tau} &= \frac{E}{1 - \frac{2M}{r}}, \\ \frac{d\phi}{d\tau} &= \frac{L}{r^2}, \\ \frac{dr}{d\tau} &= \pm \sqrt{E^2 - \left(1 - \frac{2M}{r}\right) \left(1 + \frac{L^2}{r^2}\right)}.\end{aligned}$$

where: - E is the specific energy of the test particle, - L is the specific angular momentum of the test particle.

3 Project Goals

The goal of this project is to develop a Python library that facilitates the analysis of particle motion around black holes. Specifically, the library aims to provide functionalities for calculating the effective potential, generating orbits, computing trajectories, and creating plots based on user-provided parameters. The scope of this project is limited to calculating the trajectories of radius bound Orbits, type 4 orbits described above.

3.1 Key Features

The Python library will include the following key features:

- **Effective Potential Class:** Given the angular momentum (L) of a particle, the library will calculate the effective potential (V_{eff}) using the Schwarzschild metric. This potential describes the effective energy landscape that determines the motion of the particle in the gravitational field of the black hole. Given the total energy E , the class will calculate the intersection points between the effective energy and the effective potential, these will represent the radii of possible orbits.
- **Orbit Class:** Using the calculated effective potential, the library will generate orbits for particles with specified initial conditions, the L , E , r_{\min} , r_{\max} , and the V_{eff} .
- **Trajectory Class:** The library will provide functions to compute the trajectories of particles based on the Orbit object provided. This will allow users to simulate the motion of particles and study their behavior over time.
- **Vizualization Class:** To visualize the results of the analysis, the library will include plotting utilities for generating various types of plots, such as effective potential curves, and trajectory plots. These plots will help users understand the dynamics of particle motion around black holes.

3.2 Target Users

The Python library targets those interested in studying the behavior of particles in the vicinity of black holes by providing easy-to-use functionalities for analyzing black hole orbits.

4 Architecture

This class diagram represents a system of classes and their relationships in a Python program, the simulation of trajectories and associated data. In summary, this diagram outlines a system for creating, manipulating, and visualizing trajectories and associated data, including effective potentials and orbit sampling. Classes are organized hierarchically with inheritance relationships, and there are factory classes for creating instances of certain types of objects. The visualization classes provide methods for creating plots of trajectories and effective potentials.

4.1 UML Class Diagram

4.2 Interface and Driver Integration

4.2.1 Interface Overview

The interface allows for user interaction with the Black Hole Orbital Trajectory Calculator. It allows for the input of parameters like angular momentum (L) and energy (E) and graph type and provides visualization of the effective potential and orbital paths. Users can input YAML files to set these parameters, including the specific types of orbits to be analyzed.

4.2.2 Driver Functionality

The driver reads from the YAML configuration files to extract simulation parameters such as L , E , and the orbit type, or it takes in these 3 arguments from the command line. It ensures that the library receives the correct inputs and that the results are accurately displayed through the interface. It uses the factory classes to create instances of the Effective Potential, Orbit, and Trajectory objects. As well as calls for the creation of Visualization code. The driver shows how these different features interact with each other for the execution of our goal.

5 Lessons Learned and Issues Encountered

5.1 Effective Potential Class

While working on the V_{eff} class the main challenges came during calculating the turning points for a given E . The code depends on using the scipy library, specifically the fsolve method from the optimize class. The root-finding algorithm iteratively searches for the roots of the equation within a predefined range of initial guesses, since the initial guesses are a linear space of closely separated r values, I needed to check for uniqueness of the turning points. Another issue is that the algorithm could have a runtime error when looking for a turning point when the radius falls towards infinity. The solution is that the maximum number of turning points from any combination of L and E is 3 roots, but for some combinations it is less, so I had to add if statements to check for number of roots and break from the algorithm when the correct number was reached. Some other challenges was calculating the E_u and E_s when there is no local maxima or minima, for low values of L the algorithm returns that it is not a number, therefore I had to check L values to be large enough to have inflection points.

5.2 Orbit Class

The main challenge of creating the Orbit objects was the implementation of the factory. There are 4 different categories of orbits and we are only focusign on category 4 for the scope of this project. To create an orbit object we need to retrieve turning points from the Veff class, and then utilize the type 4 category to claim an r_{min} and an r_{max} depending on the number of turning points recieved form Veff.

5.3 Trajectory Class

Main challenges were understanding the theoretical background behind the calculations of the trajectory and then applying the correct scipy library methods for implementation.

5.4 Visualizer Class

A challenge for the vizualizer plot is understanding the correct scales needed to plot the data. The r values for the Veff plot need to be in log scale to get the expected image.

5.5 Driver Code

The forming and mapping out of the relationships between the different classes was the main challenge of this code section. Making sure that the correct parameters are used in the creation of the objects and separating the jobs accordingly.

6 Hopes for Further Improvement

Further improvements would be to including an implementation for categories 1, 2, and 3 in the OrbitFactory class. Currently we do not have an algorithm for calculating orbits that have radius that plummets to 0 or towards infinity.

Another hope is to add more functionality to the vizualizer, adding animations.