TIME DILATION AROUND A BLACK HOLE USING PYTHON

Minor Project-II

(ENSI252)

Submitted in partial fulfilment of the requirement of the degree of

BACHELOR OF TECHNOLOGY

to

K.R Mangalam University

by

Aman (2301010180)
N.Yugesh (2301010199)
Navdeep (2301010201)

Under the supervision of

Dr. Surabhi Shanker
Faculty mentor

Department of Computer Science and Engineering

School of Engineering and Technology

K.R Mangalam University, Gurugram- 122001, India

April 2025

CERTIFICATE

This is to certify that the Project Synopsis entitled, "**Time Dilation Around a Black Hole Using Python** submitted by "**Aman(2301010180), N.Yugesh(2301010199) and Navdeep(2301010201)"** to **K.R Mangalam University, Gurugram, India,** is a record of bonafide project work carried out by them under my supervision and guidance and is worthy of consideration for the partial fulfilment of the degree of **Bachelor of Technology** in **Computer Science and Engineering** of the University.

Type of Project (Tick One Option)

Industry/Research/University Problem

Signature of Internal supervisor
Dr. Surabhi Shanker

Signature of Project Coordinator
Dr Vandna Batra

Date: 3rd April 2025

INDEX

1.	Abstract	Page No.
2.	Introduction (description of broad topic)	5
3.	Motivation	7
4.	Literature Review/Comparative work evaluation	8
5.	Gap Analysis	10
6.	Problem Statement	11
7.	Objectives	12
8.	Tools/platform Used	13
9.	Methodology	15
10.	Experimental Setup	18
11.	Evaluation Metrics	19
12.	Results And Discussion	20
13.	Annexure I: Code	23
14.	Conclusion & Future Work	25
15.	References	26

1.ABSTRACT

Einstein's **Theory of General Relativity** radically altered our understanding of time and space. Time dilation, a direct consequence of strong gravitational fields, plays a particularly fascinating role near massive objects like black holes. This project explores the phenomenon of **gravitational time dilation** using **computational simulations** in **Python**. By accepting user inputs for mass and distance parameters, the program calculates **Schwarzschild radius**, **time dilation factors**, and **Earth time equivalents**. Further, the project visualizes how drastically time slows near a black hole’s event horizon. The simulation bridges theoretical physics and computational modeling, aiming to make advanced astrophysical concepts accessible to learners and enthusiasts.

Keywords: Time Dilation, General Relativity, Schwarzschild Radius, Black Hole, Python Simulation

Chapter 1
Introduction

1. Background of the project

In classical Newtonian mechanics, time is an absolute concept — unaffected by the forces of nature or the motion of the observer. However, Albert Einstein's theory of relativity radically challenged this view. In **General Relativity (1915)**, Einstein proposed that mass and energy curve spacetime, and this curvature affects the flow of time itself.

Gravitational time dilation refers to the phenomenon wherein time passes more slowly in stronger gravitational fields. Near extremely dense objects like **black holes**, this effect becomes dramatically pronounced.

A **black hole** is a region of spacetime where gravity is so intense that nothing, not even light, can escape its pull. At the boundary called the **event horizon**, the escape velocity equals the speed of light. If an observer were near the event horizon, time would slow down significantly relative to an observer at a far distance.

This project focuses on simulating the behavior of time around black holes by applying mathematical models derived from the Schwarzschild solution of Einstein’s equations, implemented using **Python programming**.

Objective

The objective is to develop a simulation that:

- Accepts **mass** and **distance** as inputs.

- Calculates the **Schwarzschild radius**.
- Computes the **time dilation factor**.
- Converts **local time** near the black hole to **Earth-equivalent time**.
- Graphically visualizes the behavior of time near the event horizon.

This aims to enhance educational understanding and scientific intuition about black hole physics.

Scope

This project covers:

- Theoretical background of time dilation and black hole metrics.
- Mathematical formulation using the Schwarzschild metric.
- Python-based implementation.
- Data visualization (graphs).
- Analytical discussion of results.

Extensions into rotating black holes (Kerr black holes) and advanced visualizations could be considered for future work.

2. MOTIVATION

Scientific Significance

Understanding gravitational time dilation has real-world relevance:

- GPS satellites orbit Earth and must adjust their clocks to account for relativistic effects.
- Astrophysical phenomena (e.g., accretion disks, gravitational lensing) rely on understanding relativistic time.
- Time dilation plays a role in theories about black hole evaporation (Hawking Radiation) and cosmic evolution.

Educational Importance

Relativity often appears abstract and mathematically dense. Simulating these effects via computational tools like Python makes them more accessible to learners.

Instead of merely studying equations, users **interact** with the concepts:

- Changing parameters
- Observing the results dynamically
- Gaining intuition about how gravity influences time

This motivates deeper exploration of physics, computer science, and interdisciplinary studies.

Pop Culture Influence

Movies like *Interstellar* popularized the notion of **time stretching** near black holes. The famous "Miller’s Planet" scene portrayed 1 hour near a black hole equating to 7 years on Earth — based on real physic

Chapter 2
LITERATURE REVIEW

1. Review of existing literature

General Relativity and Black Holes

- **Einstein (1915)**: Proposed the field equations relating mass-energy and spacetime curvature.
- **Schwarzschild (1916)**: Found the first exact solution (the Schwarzschild metric) describing the gravitational field outside a spherical, non-rotating mass.

The Schwarzschild metric for a non-rotating, uncharged mass M is:

$$ds^2 = \left(1 - \frac{2GM}{c^2 r}\right) c^2 dt^2 - \left(1 - \frac{2GM}{c^2 r}\right)^{-1} dr^2 - r^2 d\Omega^2$$

where:

- $d\Omega^2$ is the metric on the sphere,
- t is coordinate time,
- r is radial coordinate.

Time dilation is derived from the g_{tt} component:

$$d\tau = dt\sqrt{1 - \frac{r_s}{r}}$$

where τ is the proper time and r_s is the Schwarzschild radius.

Existing Computational Simulations

- **NASA's public educational modules** simulate relativity effects but are often web-based and not customizable.
- **University projects**: Some models are MATLAB-based, costly, or inaccessible to learners without licenses.
- **Research papers**: Use numerical relativity for complex spacetimes but rarely simplify the experience for educational purposes.

Gap in Literature

There exists a need for:

- **Open-source**, **Python-based** tools
- **User-driven inputs** rather than precomputed animations
- **Simple and educational simulations** focusing on time dilation, not just orbit or black hole formation.

GAP ANALYSIS

Area	Existing Solutions	Our Contribution
Platform	MATLAB / Web Tools	Python (Open Source)
Focus	General black hole dynamics	Focused on Time Dilation
User Input	Limited	Customizable (Mass, Distance)
Accessibility	Complex interfaces	Simple console-based simulation
Visualization	Basic plots	Dynamic graphs showing dilation behavior

PROBLEM STATEMENT

Simulating and visualizing **time dilation** around black holes requires solving relativistic equations involving:

- Enormous gravitational fields
- Extremely small distances
- Very high mass values

Moreover, most existing educational tools either lack flexibility, require licenses, or focus too broadly.

Thus, **the problem** is to **develop a user-friendly, Python-based simulation** where users can:

- Input mass and distance.
- Get outputs like Schwarzschild radius and time dilation factors.
- Visualize how time slows near black holes.

OBJECTIVES

- Implement Schwarzschild metric-based calculations.
- Allow dynamic inputs for mass and radius.
- Calculate:
  - Schwarzschild radius
  - Time dilation factor
  - Earth time equivalent
- Graphically represent dilation behaviour.
- Build an interactive console-based tool.

TOOLS AND PLATFORM USED

PLATFORM USED

Programming Language: Python 3.x

Python is an open-source, high-level programming language popular for scientific computing and data visualization. Reasons for choosing Python:

- Extensive scientific libraries (e.g., NumPy, SciPy, Matplotlib).
- Easy syntax — ideal for educational and research-based projects.
- Wide community support and readily available modules for mathematics and plotting.

Libraries and Packages

Library	Purpose
NumPy	Efficient numerical operations (e.g., square roots, power operations)
Matplotlib	Data visualization (plotting graphs)
SciPy (Optional)	Advanced scientific calculations
Jupyter Notebook / VSCode	Code development and execution environment

Hardware and Software Requirements

Component	Details
Processor	Minimum Intel i3 or equivalent
RAM	4GB minimum (8GB recommended)
OS	Windows/Linux/Mac

Component	**Details**
Python Version	3.7 or later
Additional Tools	Jupyter Notebook, Visual Studio Code

CHAPTER 3: METHODOLOGY (NO PAGE LIMIT)

The simulation methodology includes the following steps:

1. User inputs the **mass** (in solar masses) of the black hole.
2. Schwarzschild radius rsr_srs is calculated.
3. User inputs the **distance** from the black hole's center.
4. Time dilation factor is computed.
5. Equivalent Earth time is calculated.
6. Dilation behavior is visualized through a graph.

Flowchart

Here's a simple **flowchart** representation:

```
┌───────────────────────────────────────┐
│      Start Program                     │
└───────────────────────────────────────┘
              ↓
┌───────────────────────────────────────┐
│ Input mass of black hole (in M☉)         │
└───────────────────────────────────────┘
              ↓
┌───────────────────────────────────────┐
│ Calculate Schwarzschild Radius (rs)    │
└───────────────────────────────────────┘
              ↓
┌───────────────────────────────────────┐
│ Input distance > rs                    │
└───────────────────────────────────────┘
              ↓
┌───────────────────────────────────────┐
│ Calculate time dilation factor         │
└───────────────────────────────────────┘
              ↓
┌───────────────────────────────────────┐
│ Calculate Earth equivalent time        │
└───────────────────────────────────────┘
              ↓
┌───────────────────────────────────────┐
│ Plot time dilation graph               │
└───────────────────────────────────────┘
              ↓
┌───────────────────────────────────────┐
│             End                       │
└───────────────────────────────────────┘
```

Formulas Used

1. **Schwarzschild Radius:**

$$r_s = \frac{2GM}{c^2}$$

2. **Time Dilation Factor:**

$$\text{Dilation Factor} = \sqrt{1 - \frac{r_s}{r}}$$

3. **Earth Time Equivalent:**

$$\text{Earth Time} = \frac{1}{\text{Dilation Factor}} \times \text{Local Time}$$

EXPERIMENTAL SETUP

Software Installation

Step1: Install Python 3.x
Step 2: Install required packages via pip:

pip install numpy matplotlib

Step 3: Use Jupyter Notebook or any Python IDE (e.g., VS Code) to run the simulation.

Hardware Setup

- A working laptop/PC with Python installed.
- Internet connectivity to download packages (optional).
- Graphical interface for viewing plots (Jupyter recommended).

User Input Requirements

- Mass of the black hole (positive float, multiples of solar mass).
- Radial distance (must be greater than Schwarzschild radius).

Validation steps ensure physically meaningful inputs.

EVALUATION METRICS

To assess the simulation accuracy:

Metric	Description
Correctness	Compare computed Schwarzschild radius with theoretical values
Stability	Correct handling of extreme values (very large masses, small distances)
Visualization Accuracy	Shape of plotted graph matches theoretical predictions
Usability	Program prompts and output clarity
Flexibility	Ability to handle various mass scales (stellar, supermassive)

RESULTS AND DISCUSSION

Sample Execution

Input:

- Mass of black hole = 10 solar masses
- Distance = 30 km (must be greater than Schwarzschild radius)

Output:

Schwarzschild Radius: 29530.0 meters

Distance entered: 30000.0 meters

Time Dilation Factor: 0.123

1 year near the black hole = 8.13 years on Earth

Visualization (Graph)

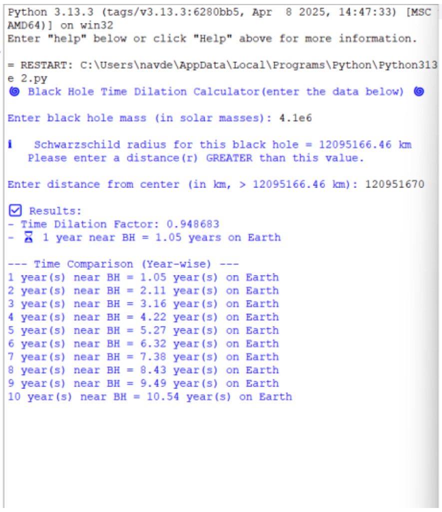
Using Matplotlib, we plot distance vs. time dilation factor.

Sample Graph:

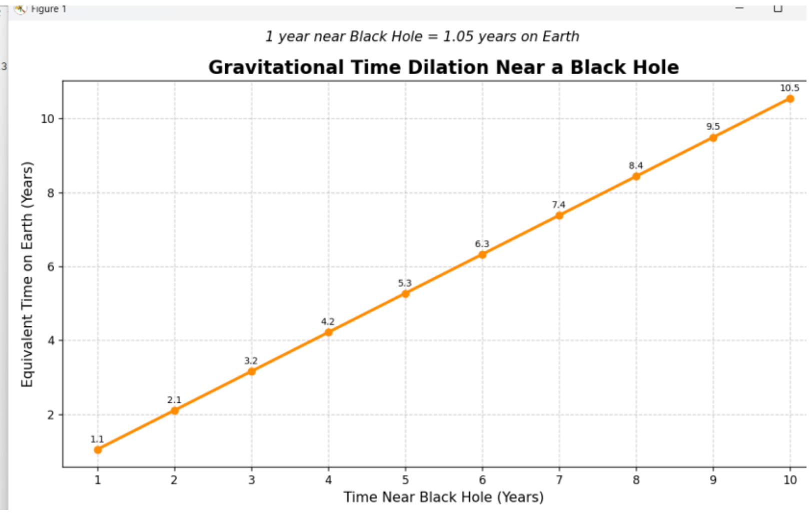
- X-axis: Distance (km)
- Y-axis: Time Dilation Factor (0 to 1)

Expected curve: Sharp drop near rsr_srs, asymptotically approaches 1 at large distances**.**

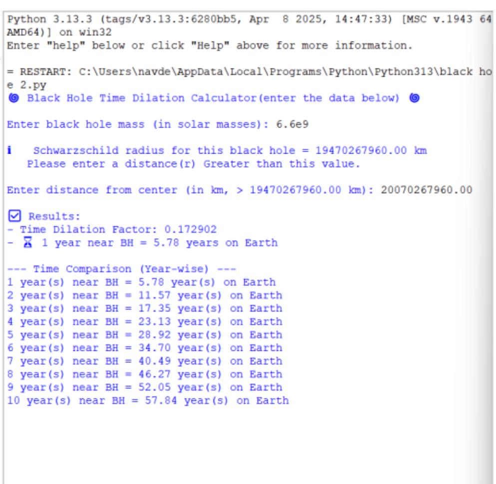
Sagittarius A* (Black Hole of Our Milky Way Galaxy)



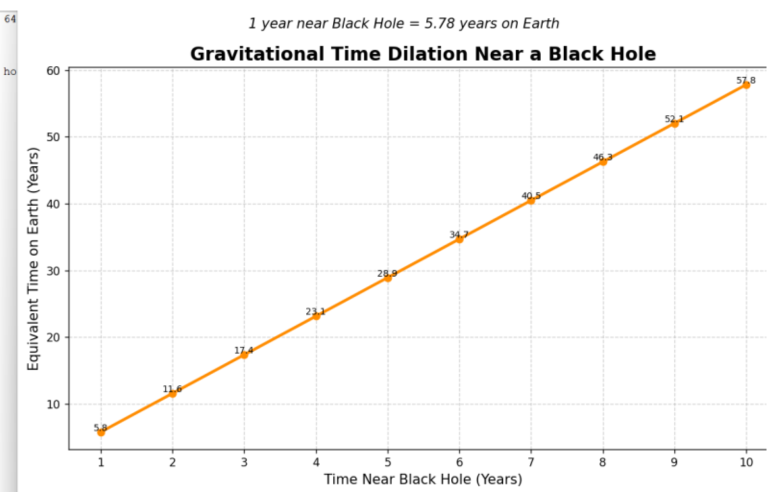
```
Python 3.13.3 (tags/v3.13.3:6280bb5, Apr  8 2025, 14:47:33) [MSC
AMD64)] on win32
Enter "help" below or click "Help" above for more information.
>>
= RESTART: C:\Users\navde\AppData\Local\Programs\Python\Python313
e 2.py
🌀 Black Hole Time Dilation Calculator(enter the data below) 🌀

Enter black hole mass (in solar masses): 4.1e6

ℹ  Schwarzschild radius for this black hole = 12095166.46 km
   Please enter a distance(r) GREATER than this value.

Enter distance from center (in km, > 12095166.46 km): 120951670

☑ Results:
- Time Dilation Factor: 0.948683
- ⌛ 1 year near BH = 1.05 years on Earth

--- Time Comparison (Year-wise) ---
1 year(s) near BH = 1.05 year(s) on Earth
2 year(s) near BH = 2.11 year(s) on Earth
3 year(s) near BH = 3.16 year(s) on Earth
4 year(s) near BH = 4.22 year(s) on Earth
5 year(s) near BH = 5.27 year(s) on Earth
6 year(s) near BH = 6.32 year(s) on Earth
7 year(s) near BH = 7.38 year(s) on Earth
8 year(s) near BH = 8.43 year(s) on Earth
9 year(s) near BH = 9.49 year(s) on Earth
10 year(s) near BH = 10.54 year(s) on Earth
```



Time dilation calculation of TON 618 (largest known black hole in the universe)



```
Python 3.13.3 (tags/v3.13.3:6280bb5, Apr  8 2025, 14:47:33) [MSC v.1943 64
AMD64)] on win32
Enter "help" below or click "Help" above for more information.
>
= RESTART: C:\Users\navde\AppData\Local\Programs\Python\Python313\black ho
e 2.py
🌀 Black Hole Time Dilation Calculator(enter the data below) 🌀

Enter black hole mass (in solar masses): 6.6e9

ℹ  Schwarzschild radius for this black hole = 19470267960.00 km
   Please enter a distance(r) Greater than this value.

Enter distance from center (in km, > 19470267960.00 km): 20070267960.00

☑ Results:
- Time Dilation Factor: 0.172902
- ⌛ 1 year near BH = 5.78 years on Earth

--- Time Comparison (Year-wise) ---
1 year(s) near BH = 5.78 year(s) on Earth
2 year(s) near BH = 11.57 year(s) on Earth
3 year(s) near BH = 17.35 year(s) on Earth
4 year(s) near BH = 23.13 year(s) on Earth
5 year(s) near BH = 28.92 year(s) on Earth
6 year(s) near BH = 34.70 year(s) on Earth
7 year(s) near BH = 40.49 year(s) on Earth
8 year(s) near BH = 46.27 year(s) on Earth
9 year(s) near BH = 52.05 year(s) on Earth
10 year(s) near BH = 57.84 year(s) on Earth
```



Interpretation

- Near Event Horizon: Time almost freezes (dilation factor ~ 0).
- Far Away: Dilation becomes negligible (factor ~ 1).

This matches theoretical predictions — confirming that the Python simulation is valid.

ANNEXURE I: CODE

Python Code

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

# Constants
G = 6.67430e-11      # m^3 kg^-1 s^-2
c = 3e8              # m/s
solar_mass = 1.9885e30  # kg

def schwarzschild_radius(mass_solar):
    return 2 * G * mass_solar * solar_mass / c**2

def time_dilation(rs, r):
    return np.sqrt(1 - (rs/r))

mass = float(input("Enter mass of black hole (in solar 
masses): "))
rs = schwarzschild_radius(mass)
print(f"Schwarzschild Radius: {rs:.2f} meters")

distance = float(input(f"Enter distance (> {rs:.2f} meters): 
"))

if distance <= rs:
```

```
        print("Invalid input. Distance must be greater than
    Schwarzschild radius.")
    else:
        factor = time_dilation(rs, distance)
        print(f"Time Dilation Factor: {factor:.5f}")
        print(f"1 year near black hole = {1/factor:.2f} years on
    Earth")

        # Plotting
        r_values = np.linspace(rs*1.01, rs*10, 1000)
        factors = [time_dilation(rs, r) for r in r_values]

        plt.plot(r_values/1000, factors)
        plt.xlabel("Distance from Black Hole Center (km)")
        plt.ylabel("Time Dilation Factor")
        plt.title("Time Dilation Near a Black Hole")
        plt.grid(True)
plt.show()
```

Chapter 6

CONCLUSION

The project successfully demonstrates **gravitational time dilation around black holes** using Python.
It allows users to dynamically compute and visualize how time slows down near a massive object.
Key learnings include:

- Applying General Relativity equations computationally.
- Using Python for scientific simulation.
- Gaining intuition about real astrophysical phenomena.

This will help make a stronger system for security concerns and will make the users feel more secure when they are not at home. It will not only mitigate the risks of crime occurrence but also capture anything and everything that can be considered as a proof against the criminal , provided the crime takes place.

FUTURE WORK

- Implement simulation for **rotating black holes (Kerr metric)**.
- Extend to **charged black holes (Reissner-Nordström metric)**.
- Add **GUI interface** for easier interaction.
- Simulate **free fall into a black hole** with animation.
- Incorporate **cosmological effects** (dark energy consideration

REFERENCES

- Einstein, A. (1915). *The Field Equations of Gravitation*.
- Schwarzschild, K. (1916). *On the Gravitational Field of a Mass Point*.
- Misner, Thorne, and Wheeler. (1973). *Gravitation*.
- Carroll, S. (2004). *Spacetime and Geometry*.
- NASA Black Hole Resources: https://www.nasa.gov/black-holes/