General Relativity Simulator

Boxiong Qiao

December 10, 2025

1 Project Description

This project is a GR engine designed to simulate and visualize the geometry of Kerr spacetime (rotating black holes). It calculates the paths (geodesics) of both massive particles and photon, and draws 2d trajectory, 3d trajectory, or an embedding diagram. It also performs gravitational lensing to render realistic images of a black hole's shadow against a star field, or simply against a grid background to see the distortion of lightfield.

The project consists of four main components:

1. physics/kerr_dynamics.py

This program implements all the physics you need in this problem. It uses Mino time in Boyer-Lindquist coordinates to fully decouple the geodesic equation, and take full advantage of all three conserved quantities making the equation easier and more stable. The equation used in the program is a set of 4 second order ODEs, with 2 of them integrable. It produces two important functions: _core_timelike and _core_null. These functions take a vector of current state, and produce (second) derivatives for the integrator.

2. numeric/integrator.py

This program contains a RKF45 integrator (Runge–Kutta–Fehlberg methodof 4(5)th order). it can solve a set of first order ODEs with high accuracy and fast speed. The main feature is that it can self-adjust step length by estimating errors on each step and can thus control the overall error of the function.

3. visualizer.py

This program can take the trajectory of a particle in Boyer-Lindquist coordinates, and can draw a 3D embedding diagram(2D slice of the equator of the black hole embedded into a 3D Euclidean space), a 2D top down view, and a 3D trajectory.

4. main.py

This program utilizes the three above programs, and takes input parameters from the command line to simulate different configurations. It generally just combines the three above programs. It also contains a verification module that calculates relative error by comparing a conserved quantity before and after the simulation.

5. Kerr_lensing.py

This program generates photorealistic images of a black hole. The code inside is (maybe not) polished for speed and enables parallel computation to get a relatively good performance on Pi 5. It uses a back ray tracing method and calculates a numerous amount of null geodesics to get the color for each pixel. It implements an observer frame called an LNRF that correctly calculates the geometry around the camera to produce the correct result. Finally, it either generates a grid background or reads a picture as its background (like starmap.jpg). This program does not have a argparsing part and all parameters are encoded in the program.

6. gui.py

This program combines all the above programs and creates a graphic user interface where users can easily adjust the input parameters and see the result in the same window, using PyQt. It should handles errors elegantly(hopefully).

2 Result and usage

- `main.py`: Command-line interface for orbit and scattering simulation.
- `Kerr_lensing.py`: Standalone ray-tracing script for gravitational lensing.
- `gui.py`: Full Graphical User Interface (GUI) integrating all features.

2.1 Visualization of GR Simulation: `main.py`

To run a simulation, use the following command structure. You can customize the black hole mass (M), spin (a), and particle parameters.

```
python main.py --task orbit --mode null --M 1.0 --a 0.99 \
               --lam 3.0 --eta 0.0 --r0 6.0 --t_max 100 --out orbit
```

Listing 1: Running a Null Geodesic (Photon) Simulation(Default)

```
python main.py --task orbit --mode timelike --M 1.0 --a 0.5 \
               --E 0.95 --L 2.5 --C 5.0 --r0 10.0 --t_max 500
```

Listing 2: Running a Timelike Geodesic (Particle) Simulation

Key Arguments:

`--mode`: `null` (photons) or `timelike` (particles).

`--M /--a`: mass and spin of the black hole.

`--lam / --eta`: proportional angular momentum and energy for null geodesics.

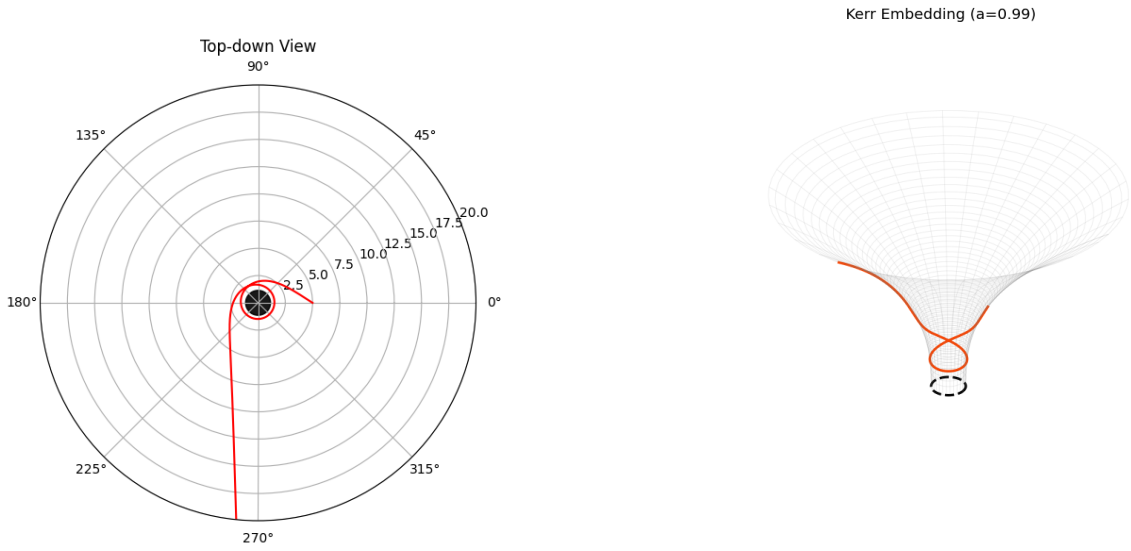
`--E / --L / --C`: energy, angular momentum and Carter constant for timelike geodesics.

`--r0`: initial distance from black hole. Note that other initial states such as angular position, velocity, are defined by the above two(three) parameters for null(timelike) geodesic.

`--out`: the prefix name of output figures.

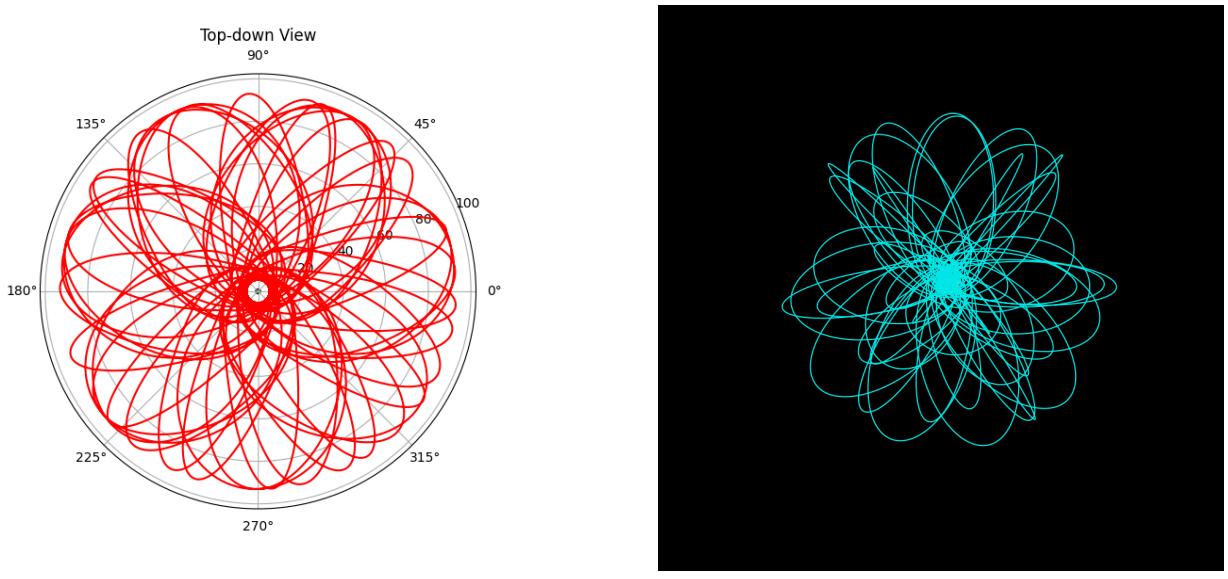
`--t_max`: maximum calculated Mino time. Note that if you see the trajectory looks like a ball of yarn, that is correct(you may shorten t_max to see more clearly)

The script outputs trajectory plots. For a trajectory that is not equatorial, the program automatically switches to 3D plots instead of the embedded diagram.



(a) Top-Down View (Polar) (b) Embedding diagram

Figure 1: Output generated by python main.py –r0 5 –lam 2.5.



(a) Top-Down View (Polar) (b) 3D Trajectory

Figure 2: timelike output generated by python main.py –mode timelike –r0 6.

2.2 Gravitational lensing: `Kerr_lensing.py`

Since parameters are currently hardcoded in the `__main__` block, simply execute the script:

```
python Kerr_lensing.py
```

Note: The script uses Numba's `prange` *for parallel processing and may take a few seconds to compile JIT kernels.*

The script generates a image mapping the distorted spacetime against a grid background and a background star field (or noise if no starmap is found).

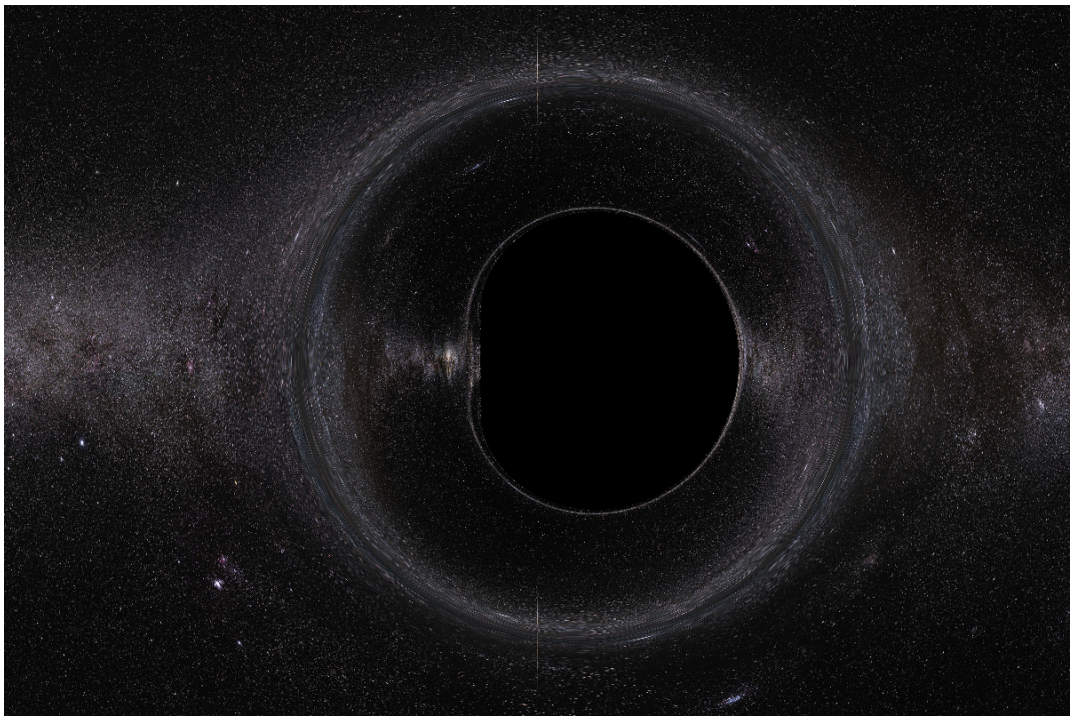


Figure 3: Kerr black hole with extreme spin. I did not show the grid background because I want to make this picture BIGGGG!

2.3 Interactive GUI: `gui.py`

Launch the application using PyQt5:

```
python gui.py
```

The "Orbit & Scattering" tab allows real-time visualization of geodesics.

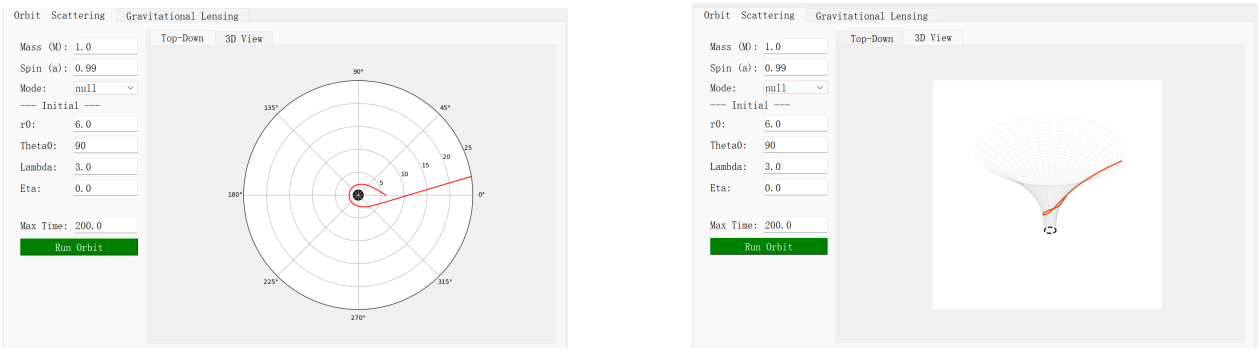


Figure 4: photon output of gui.py.

The "Gravitational Lensing" tab renders the black hole scene. It supports loading a custom `starmap.jpg`.

(a) no starmap choosen (b) starmap

Figure 5: lensing output by gui.py.

3 References

Solving Kerr geodesic:

Null and time-like geodesics in Kerr-Newman black hole exterior Chen-Yu Wang et al, http://dx.doi.org/10.1103/PhysRevD.106.084048

LNRF of the camera near a black hole:

Light escape cones in local reference frames of Kerr–de Sitter black hole spacetimes and related black hole shadows, Zdenek Stuchlik et al, Eur. Phys. J. C (2018) 78:180