Bosun Park
PHYS 161 – Black Holes
University of California San Diego
Instructor: Prof. George M. Fuller
Final Project Write-Up
December 11, 2025

Ray-Tracing Black Holes: Schwarzschild and Kerr

In this final project for PHYS 161 (Black Holes), my goal was to turn the general relativity we learned in class into concrete images of black holes. Instead of only seeing the metric and geodesic equations on paper, I wanted to build a simple ray-tracing code that uses those equations to show the black hole shadow, an accretion disk, and a background sky. I focused on two spacetimes: the non-spinning Schwarzschild black hole and the spinning Kerr black hole. By comparing images from these two metrics, I could see directly how spin, frame dragging, and relativistic Doppler effects change the appearance of a black hole. To make the light bending more obvious, I rendered images with horizontal and vertical rainbow backgrounds on a sphere, and with a realistic Milky Way all-sky map. I then added a thin orbiting disk and used a relativistic redshift factor to color it, so the disk side moving toward the camera becomes brighter and bluer, and the receding side becomes dimmer and redder.

In this project I used units $G = c = 1$and the $(- + + +)$ sign convention, matching Hartle and the PHYS 161 notes. For the non-spinning case I used the Schwarzschild metric in coordinates (t, r, θ, ϕ):

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

For the spinning case I used the Kerr metric in Boyer–Lindquist coordinates. It is written in terms of

$$\Sigma = r^2 + a^2 \cos^2\theta, \qquad \Delta = r^2 - 2Mr + a^2.$$

Then the line element is

$$ds^2 = \; -\left(1 - \frac{2Mr}{\Sigma}\right) dt^2 - \frac{4Mar\sin^2\theta}{\Sigma}\, dt\, d\phi + \frac{\Sigma}{\Delta}\, dr^2 + \Sigma\, d\theta^2 \\ + \left(r^2 + a^2 + \frac{2Ma^2 r\sin^2\theta}{\Sigma}\right) \sin^2\theta\, d\phi^2.$$

In both spacetimes, light rays follow null geodesics which satisfy

$$\frac{d^2x^\mu}{d\lambda^2} + \Gamma^\mu_{\alpha\beta} \frac{dx^\alpha}{d\lambda} \frac{dx^\beta}{d\lambda} = 0,$$

together with the null condition

$$g_{\mu\nu} k^\mu k^\nu = 0, \qquad k^\mu = \frac{dx^\mu}{d\lambda}.$$

For the disk gas and the observer, I used four-velocities u^μ normalized by

$$g_{\mu\nu} u^\mu u^\nu = -1.$$

The relativistic redshift/Doppler factor relating emitted and observed frequency is

$$g = \frac{\nu_{\mathrm{obs}}}{\nu_{\mathrm{emit}}} = \frac{-(u_{\mathrm{obs}} \cdot k)}{-(u_{\mathrm{emit}} \cdot k)}, \qquad u \cdot k = g_{\mu\nu} u^\mu k^\nu.$$

This scalar g is what I use later when coloring the disk: $g > 1$ gives blueshift/boost and $g < 1$ gives redshift/dimming.

Numerically, I organized the project into a simple 4-box pipeline. First, I choose the spacetime (Schwarzschild or Kerr) and implement the metric $g_{\mu\nu}(r,\theta)$. For Schwarzschild I hard-code analytic Christoffel symbols $\Gamma^\mu_{\alpha\beta}(r,\theta)$. For Kerr I compute finite-difference derivatives of $g_{\mu\nu}(r,\theta)$ in r and θ to build approximate Christoffels. Second, I place a camera at radius $r = R_{\mathrm{cam}}$ and build a local orthonormal tetrad; for each pixel on the image plane, I construct an initial photon four-momentum k^μ corresponding to a given direction in the camera frame and transform it into coordinate components. Third, I integrate the geodesic equations as a first-order system for the state

$$y = (t, r, \theta, \phi, u^t, u^r, u^\theta, u^\phi)$$

using [solve_ivp] (RK45). During integration I check for three outcomes: (1) r falls inside the horizon radius r_+, (2) the photon crosses the equatorial plane $\theta = \pi/2$ within the disk's radial range, or (3) r grows beyond a large cutoff radius, which I treat as "escaped to the sky." Finally, I classify the ray and assign a pixel color: horizon hits are black, disk hits use a base disk color modified by g, and escaping rays are mapped onto either a rainbow sphere or the Milky Way EXR sky using the final direction (θ, ϕ).

All of this is implemented in Python using my [nonSpinningSchwarzschildDopplerExr.py] and [SpinningKerrDopplerExr.py] codes, running inside a Conda environment I call "grtracer". The overall structure is inspired by GRTracer-style ray-tracing codes: define a metric, compute the Christoffels, integrate null geodesics, then convert the geodesic endpoints into pixel colors.

With a horizontal rainbow background, the Schwarzschild black hole produces a symmetric dark circular shadow in the center, with horizontal color bands smoothly bent around it. The upper and lower stripes wrap around the shadow and meet, clearly showing gravitational lensing. For Kerr with nonzero spin a, the shadow becomes asymmetric and appears shifted to one side, and the bending of the stripes is no longer perfectly symmetric. When I add a thin orbiting disk, the side of the ring where the gas orbits toward the camera is much brighter and slightly bluer (because $g > 1$there), while the opposite side is dimmer and redder ($g < 1$).

With a vertical rainbow background, Schwarzschild bends vertical stripes into curves around the shadow, and Kerr again shows a noticeable left–right asymmetry and uneven brightness. Using a realistic Milky Way all-sky map as the background, the shadow sits in front of the Galactic plane, and the stars and dust lanes are strongly distorted around it. Adding the Doppler-boosted disk on top reproduces the familiar bright crescent around the black hole.

This project helped me connect the mathematical side of general relativity to actual visual intuition. Working directly with the metric, Christoffel symbols, and the geodesic equation made me see how changing from Schwarzschild to Kerr really changes light paths and therefore the shape and position of the shadow. Implementing the redshift factor g from four-velocities and photon momenta also made relativistic Doppler beaming feel more concrete, because I could see one side of the disk brighten and the other side fade.