Practical III
Visualizing Event Horizon and Ergosphere (Singularities) of Kerr Metric or Black Hole

Gaurav Bhoir
UID No: 2309103

Aim

To visualize the event horizon and ergosphere of Kerr black holes with different spin parameters using the Kerr metric.

Theory

The Kerr metric describes the geometry of spacetime around a rotating black hole. Unlike the non-rotating Schwarzschild black hole, the Kerr black hole has two distinct horizons: the event horizon and the ergosphere. The event horizon is the boundary beyond which nothing can escape, while the ergosphere is the region outside the event horizon where objects cannot remain in place due to frame-dragging effects.

Kerr Metric

$$ds^2 = -\left(1 - \frac{2GMr}{c^2\Sigma}\right)c^2dt^2 + \frac{\Sigma}{\Delta}dr^2 + \Sigma d\theta^2 + \left(r^2 + a^2 + \frac{2GMa^2r\sin^2\theta}{c^2\Sigma}\right)\sin^2\theta d\phi^2 - \frac{4GMar\sin^2\theta}{c\Sigma}cdtd\phi \tag{1}$$

where

$$\Sigma = r^2 + a^2\cos^2\theta \tag{2}$$

$$\Delta = r^2 - \frac{2GMr}{c^2} + a^2 \tag{3}$$

Event Horizons

$$r_\pm = \frac{GM}{c^2} \pm \sqrt{\left(\frac{GM}{c^2}\right)^2 - a^2} \tag{4}$$

Ergosphere

$$r_{\text{ergo}} = \frac{GM}{c^2} \pm \sqrt{\left(\frac{GM}{c^2}\right)^2 - a^2\cos^2\theta} \tag{5}$$

Procedure

1. Import Required Libraries
   - Import `numpy` for numerical operations.
   - Import `astropy.units` for handling units.
   - Import `matplotlib.pyplot` for plotting.
   - Import relevant modules from `einsteinpy` for Kerr metric calculations.
2. Define Initial Parameters
   - Set mass M of the black hole.
   - Set two different spin parameters a_1 and a_2 for the Kerr black holes.

3. Initialize Coordinates
    - Define a Boyer-Lindquist coordinate object to initialize the metric.
4. Define Kerr Black Holes
    - Create Kerr black holes with the defined mass and spin parameters.
5. Obtain Singularities
    - Extract the inner and outer horizons and ergospheres from the Kerr metric.
6. Prepare Data for Plotting
    - Sample polar angles for plotting.
    - Calculate the coordinates of the ergospheres and event horizons for both black holes.
7. Plotting
    - Create a figure with two subplots.
    - Plot the inner and outer ergospheres and horizons for both black holes.

Python Code

```python
import astropy.units as u
import numpy as np
import matplotlib.pyplot as plt
from einsteinpy.coordinates import BoyerLindquistDifferential
from einsteinpy.metric import Kerr
M = 10e30 * u.kg
a1 = 0.4 * u.one
a2 = 0.9 * u.one # Extremal Kerr Black Hole
bl =
    BoyerLindquistDifferential(t=0.*u.s,r=1e3*u.m,theta=np.pi/2*u.rad,phi=np.pi*u.rad,v_r=0.*u.m/.s,v

kerr1 = Kerr(coords=bl, M=M, a=a1)
kerr2 = Kerr(coords=bl, M=M, a=a2)

sing_dict1 = kerr1.singularities()
sing_dict2 = kerr2.singularities()
print(sing_dict1, sing_dict2, sep="\n\n")

theta = np.linspace(0, 2 * np.pi, 100)
Ei1, Eo1 = sing_dict1["inner_ergosphere"], sing_dict1["outer_ergosphere"]
Ei2, Eo2 = sing_dict2["inner_ergosphere"], sing_dict2["outer_ergosphere"]
Ei1_list, Eo1_list = Ei1(theta), Eo1(theta)
Ei2_list, Eo2_list = Ei2(theta), Eo2(theta)
# For Black Hole 1 (a = 0.4)
Xei1 = Ei1_list * np.sin(theta)
Yei1 = Ei1_list * np.cos(theta)
Xeo1 = Eo1_list * np.sin(theta)
Yeo1 = Eo1_list * np.cos(theta)
# For Black Hole 2 (a = 0.9)
Xei2 = Ei2_list * np.sin(theta)
Yei2 = Ei2_list * np.cos(theta)
Xeo2 = Eo2_list * np.sin(theta)
Yeo2 = Eo2_list * np.cos(theta)
# Event Horizons
Hi1, Ho1 = sing_dict1["inner_horizon"], sing_dict1["outer_horizon"]
Hi2, Ho2 = sing_dict2["inner_horizon"], sing_dict2["outer_horizon"]
# For Black Hole 1 (a = 0.4)
Xhi1 = Hi1 * np.sin(theta)
Yhi1 = Hi1 * np.cos(theta)
Xho1 = Ho1 * np.sin(theta)
```

```
Yho1 = Ho1 * np.cos(theta)
# For Black Hole 2 (a = 0.9)
Xhi2 = Hi2 * np.sin(theta)
Yhi2 = Hi2 * np.cos(theta)
Xho2 = Ho2 * np.sin(theta)
Yho2 = Ho2 * np.cos(theta)

fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(15, 7.5))
ax1.fill(Xei1, Yei1, 'b', Xeo1, Yeo1, 'r', Xhi1, Yhi1, 'b', Xho1, Yho1, 'r', alpha=0.3)
ax1.set_title(f"$M = {M}, a = {a1}$", fontsize=18)
ax1.set_xlabel("X", fontsize=18)
ax1.set_ylabel("Y", fontsize=18)
ax1.set_xlim([-6100, 6100])
ax1.set_ylim([-6100, 6100])

ax2.fill(Xei2, Yei2, 'b', Xeo2, Yeo2, 'r', Xhi2, Yhi2, 'b', Xho2, Yho2, 'r', alpha=0.3)
ax2.set_title(f"$M = {M}, a = {a2}$", fontsize=18)
ax2.set_xlabel("X", fontsize=18)
ax2.set_ylabel("Y", fontsize=18)
ax2.set_xlim([-6100, 6100])
ax2.set_ylim([-6100, 6100])

position = [2.5, np.pi / 2, 0.]
momentum = [0., 0., -2.]
a = 0.99
steps = 7440  # As close as we can get before the integration becomes highly unstable
delta = 0.0005
omega = 0.01
suppress_warnings = True
geod =
    Nulllike(metric="Kerr",metric_params=(a,),position=position,momentum=momentum,steps=steps,delta=de
sgpl = StaticGeodesicPlotter(bh_colors=("red", "blue"))
sgpl.plot2D(geod, coordinates=(1, 2), figsize=(10, 10), color="indigo")  # Plot X vs Y
sgpl.show()
```

Output

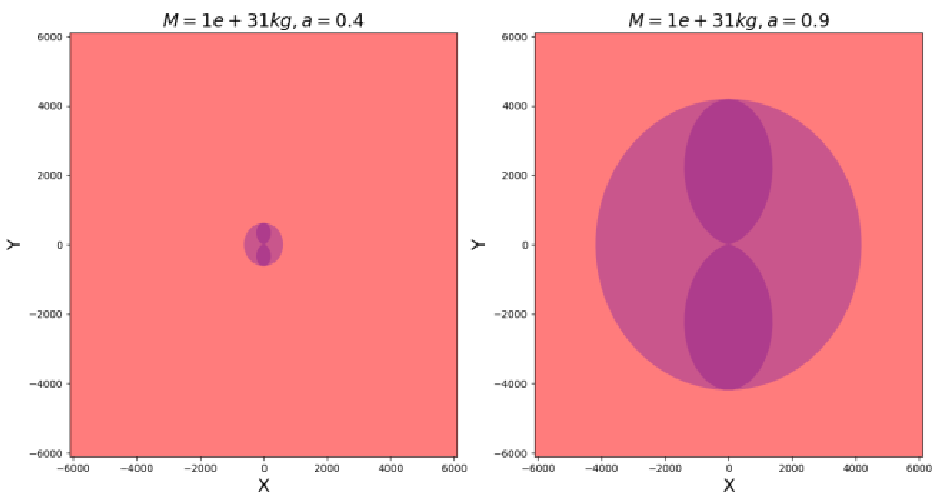

Figure 1:
Left represents a Kerr black hole with spin parameter a=0.4, The inner regions represent the inner and outer event horizons. The red regions represent the inner and outer ergospheres.

The inner regions represent the inner and outer event horizons. The red regions represent the inner and outer ergospheres.

```
'inner horizon': 619.9719578966824, 'outer horizon': 14232.348580340647
'inner horizon': 4189.172053956158, 'outer horizon': 10663.148484281173
'inner horizon': 619.9719578966824, 'outer horizon': 14232.348580340647
'inner horizon': 4189.172053956158, 'outer horizon': 10663.148484281173
```

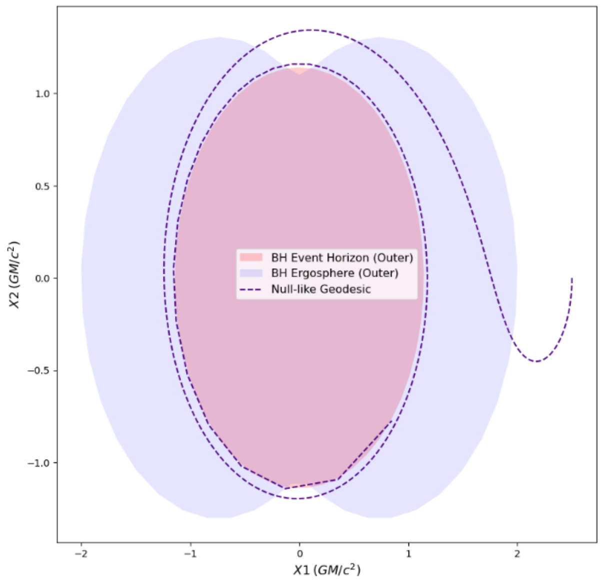

Figure 1: 2D plot showing the trajectory of a photon around an extremal Kerr black hole, illustrating the complex path due to the black hole's intense gravitational field and high spin.

Conclusion

By visualizing the event horizons and ergospheres of Kerr black holes, we can observe the following:

- For higher spin parameters, the individual singularities (ergospheres and event horizons) become more distinct.
- As the spin parameter approaches zero, the singularities converge into a single surface, representing the event horizon of a Schwarzschild black hole.