GTR Project

Samarth Bhatnagar(2023B5A70853G)

May 2025

1 Introduction

The spacetime metric tensor $g_{\mu\nu}$, contains the information on how the spacetime is curved, and here we will find the Christoffel coefficients, geodesic equations, Ricci Tensor components, Ricci Scalar and the Riemann curvature tensor.

2 For a general diagonal $g_{\mu\nu}$ with rotational symmetry

We have considered the following $g_{\mu\nu}$:

$$g_{\mu\nu} = \begin{pmatrix} A & 0 & 0 & 0 \\ 0 & -B & & 0 \\ 0 & 0 & -r^2 & 0 \\ 0 & 0 & 0 & -r^2 sin^2(\theta) \end{pmatrix}$$

and we get the following results:

2.1 Non-zero Christoffel Symbols

$$\Gamma^{0}_{01} = \frac{\frac{d}{dr}A(r)}{2A(r)}$$

$$\Gamma^{0}_{10} = \frac{\frac{d}{dr}A(r)}{2A(r)}$$

$$\Gamma^{1}_{00} = \frac{\frac{d}{dr}A(r)}{2B(r)}$$

$$\Gamma^{1}_{11} = \frac{\frac{d}{dr}B(r)}{2B(r)}$$

$$\Gamma^{1}_{22} = -\frac{r}{B(r)}$$

$$\Gamma^{1}_{33} = -\frac{r\sin^{2}(\theta)}{B(r)}$$

$$\Gamma^{2}_{12} = \frac{1}{r}$$

$$\Gamma^{2}_{21} = \frac{1}{r}$$

$$\Gamma^{2}_{33} = -\frac{\sin(2\theta)}{2}$$

$$\Gamma^{3}_{13} = \frac{1}{r}$$

$$\Gamma^{3}_{23} = \frac{1}{\tan(\theta)}$$

$$\Gamma^{3}_{31} = \frac{1}{r}$$

$$\Gamma^{3}_{32} = \frac{1}{\tan(\theta)}$$

2.2 Geodesic Equations

$$
\begin{aligned}
\frac{d^2t}{ds^2} &= -\frac{v_0 v_1 \frac{d}{dr}A(r)}{A(r)}\\
\frac{d^2r}{ds^2} &= \frac{r v_2^2 + r v_3^2 \sin^2(\theta) - \frac{v_0^2 \frac{d}{dr}A(r)}{2} - \frac{v_1^2 \frac{d}{dr}B(r)}{2}}{B(r)}\\
\frac{d^2\theta}{ds^2} &= \frac{v_3^2 \sin(2\theta)}{2} - \frac{2 v_1 v_2}{r}\\
\frac{d^2\phi}{ds^2} &= -\frac{2 v_3 \left(\frac{r v_2}{\tan(\theta)} + v_1\right)}{r}
\end{aligned}
$$

**here, v_0, v_1, v_2, v_3 are $\frac{dt}{ds}, \frac{dr}{ds}, \frac{d\theta}{ds}, \frac{d\phi}{ds}$ respectively.

2.3 Non-zero Ricci Tensor Components

$$
\begin{aligned}
R_{00} &= \frac{\frac{d^2}{dr^2}A(r)}{2B(r)} - \frac{\frac{d}{dr}A(r)\frac{d}{dr}B(r)}{4B^2(r)} - \frac{\left(\frac{d}{dr}A(r)\right)^2}{4A(r)B(r)} + \frac{\frac{d}{dr}A(r)}{rB(r)}\\
R_{11} &= -\frac{\frac{d^2}{dr^2}A(r)}{2A(r)} + \frac{\frac{d}{dr}A(r)\frac{d}{dr}B(r)}{4A(r)B(r)} + \frac{\left(\frac{d}{dr}A(r)\right)^2}{4A^2(r)} + \frac{\frac{d}{dr}B(r)}{rB(r)}\\
R_{22} &= \frac{r\frac{d}{dr}B(r)}{2B^2(r)} - \frac{r\frac{d}{dr}A(r)}{2A(r)B(r)} + 1 - \frac{1}{B(r)}\\
R_{33} &= \frac{\left(rA(r)\frac{d}{dr}B(r) - rB(r)\frac{d}{dr}A(r) + 2\left(B(r)-1\right)A(r)B(r)\right)\sin^2(\theta)}{2A(r)B^2(r)}
\end{aligned}
$$

2.4 Ricci Scalar

$$
R = \frac{\frac{d^2}{dr^2}A(r)}{A(r)B(r)} - \frac{\frac{d}{dr}A(r)\frac{d}{dr}B(r)}{2A(r)B^2(r)} - \frac{\left(\frac{d}{dr}A(r)\right)^2}{2A^2(r)B(r)} - \frac{2\frac{d}{dr}B(r)}{rB^2(r)} + \frac{2\frac{d}{dr}A(r)}{rA(r)B(r)} - \frac{2}{r^2} + \frac{2}{r^2B(r)}
$$

2.5 Non-zero Riemann Tensor Components

$$R^{0}_{101} = -\frac{\frac{d^2}{dr^2}A(r)}{2A(r)} + \frac{\frac{d}{dr}A(r)\frac{d}{dr}B(r)}{4A(r)B(r)} + \frac{\left(\frac{d}{dr}A(r)\right)^2}{4A^2(r)}$$

$$R^{0}_{110} = \frac{\frac{d^2}{dr^2}A(r)}{2A(r)} - \frac{\frac{d}{dr}A(r)\frac{d}{dr}B(r)}{4A(r)B(r)} - \frac{\left(\frac{d}{dr}A(r)\right)^2}{4A^2(r)}$$

$$R^{0}_{202} = -\frac{r\frac{d}{dr}A(r)}{2A(r)B(r)}$$

$$R^{0}_{220} = \frac{r\frac{d}{dr}A(r)}{2A(r)B(r)}$$

$$R^{0}_{303} = -\frac{r\sin^2(\theta)\frac{d}{dr}A(r)}{2A(r)B(r)}$$

$$R^{0}_{330} = \frac{r\sin^2(\theta)\frac{d}{dr}A(r)}{2A(r)B(r)}$$

$$R^{1}_{001} = -\frac{\frac{d^2}{dr^2}A(r)}{2B(r)} + \frac{\frac{d}{dr}A(r)\frac{d}{dr}B(r)}{4B^2(r)} + \frac{\left(\frac{d}{dr}A(r)\right)^2}{4A(r)B(r)}$$

$$R^{1}_{010} = \frac{\frac{d^2}{dr^2}A(r)}{2B(r)} - \frac{\frac{d}{dr}A(r)\frac{d}{dr}B(r)}{4B^2(r)} - \frac{\left(\frac{d}{dr}A(r)\right)^2}{4A(r)B(r)}$$

$$R^{1}_{212} = \frac{r\frac{d}{dr}B(r)}{2B^2(r)}$$

$$R^{1}_{221} = -\frac{r\frac{d}{dr}B(r)}{2B^2(r)}$$

$$R^{1}_{313} = \frac{r\sin^2(\theta)\frac{d}{dr}B(r)}{2B^2(r)}$$

$$R^{1}_{331} = -\frac{r\sin^2(\theta)\frac{d}{dr}B(r)}{2B^2(r)}$$

$$R^{2}_{002} = -\frac{\frac{d}{dr}A(r)}{2rB(r)}$$

$$R^{2}_{020} = \frac{\frac{d}{dr}A(r)}{2rB(r)}$$

$$R^{2}_{112} = -\frac{\frac{d}{dr}B(r)}{2rB(r)}$$

$$R^{2}_{121} = \frac{\frac{d}{dr}B(r)}{2rB(r)}$$

$$R^{2}_{323} = \frac{(B(r)-1)\sin^2(\theta)}{B(r)}$$

$$R^{2}_{332} = \frac{(1-B(r))\sin^2(\theta)}{B(r)}$$

$$R^3_{003} = -\frac{\frac{d}{dr}A(r)}{2rB(r)}$$
$$R^3_{030} = \frac{\frac{d}{dr}A(r)}{2rB(r)}$$
$$R^3_{113} = -\frac{\frac{d}{dr}B(r)}{2rB(r)}$$
$$R^3_{131} = \frac{\frac{d}{dr}B(r)}{2rB(r)}$$
$$R^3_{223} = -1 + \frac{1}{B(r)}$$
$$R^3_{232} = 1 - \frac{1}{B(r)}$$

3 For a Schwarzschild metric

Here the $g_{\mu\nu}$ is as such:

$$g_{\mu\nu} = \begin{pmatrix} (1-\frac{2m}{c^2r}) & 0 & 0 & 0 \\ 0 & -(1-\frac{2m}{c^2r})^{-1} & & 0 \\ 0 & 0 & -r^2 & 0 \\ 0 & 0 & 0 & -r^2 sin^2(\theta) \end{pmatrix}$$

and we get the following results:

3.1 Non-zero Christoffel Symbols

$$\Gamma^{0}_{01} = \frac{m}{r\left(c^{2}r - 2m\right)}$$
$$\Gamma^{0}_{10} = \frac{m}{r\left(c^{2}Ar - 2m\right)}$$
$$\Gamma^{1}_{00} = \frac{m\left(c^{2}r - 2m\right)}{c^{2}r^{3}}$$
$$\Gamma^{1}_{11} = -\frac{m}{r\left(c^{2}r - 2m\right)}$$
$$\Gamma^{1}_{22} = -r + \frac{2m}{c^{2}}$$
$$\Gamma^{1}_{33} = \frac{\left(-c^{2}r + 2m\right)\sin^{2}\left(\theta\right)}{c^{2}}$$
$$\Gamma^{2}_{12} = \frac{1}{r}$$
$$\Gamma^{2}_{21} = \frac{1}{r}$$
$$\Gamma^{2}_{33} = -\frac{\sin\left(2\theta\right)}{2}$$
$$\Gamma^{3}_{13} = \frac{1}{r}$$
$$\Gamma^{3}_{23} = \frac{1}{\tan\left(\theta\right)}$$
$$\Gamma^{3}_{31} = \frac{1}{r}$$
$$\Gamma^{3}_{32} = \frac{1}{\tan\left(\theta\right)}$$

3.2 Geodesic Equations

$$\frac{d^{2}t}{ds^{2}} = -\frac{2mv_{0}v_{1}}{r\left(c^{2}r - 2m\right)}$$
$$\frac{d^{2}r}{ds^{2}} = \frac{c^{2}mr^{2}v_{1}^{2} - mv_{0}^{2}\left(c^{2}r - 2m\right)^{2} + r^{3}\left(v_{2}^{2} + v_{3}^{2}\sin^{2}\left(\theta\right)\right)\left(c^{2}r - 2m\right)^{2}}{c^{2}r^{3}\left(c^{2}r - 2m\right)}$$
$$\frac{d^{2}\theta}{ds^{2}} = \frac{v_{3}^{2}\sin\left(2\theta\right)}{2} - \frac{2v_{1}v_{2}}{r}$$
$$\frac{d^{2}\phi}{ds^{2}} = -\frac{2v_{3}\left(\frac{rv_{2}}{\tan\left(\theta\right)} + v_{1}\right)}{r}$$

**here, v_0, v_1, v_2, v_3 are $\frac{dt}{ds}, \frac{dr}{ds}, \frac{d\theta}{ds}, \frac{d\phi}{ds}$ respectively.

3.3 Non-zero Ricci Tensor Components

There are no non-zero components in the Ricci tensor.

3.4 Ricci Scalar

$$R = 0$$

3.5 Non-zero Riemann Tensor Components

$$R^{0}_{101} = \frac{2m}{r^{2} \left(c^{2} r - 2m\right)}$$

$$R^{0}_{110} = - \frac{2m}{r^{2} \left(c^{2} r - 2m\right)}$$

$$R^{0}_{202} = - \frac{m}{c^{2} r}$$

$$R^{0}_{220} = \frac{m}{c^{2} r}$$

$$R^{0}_{303} = - \frac{m \sin^{2}{\left(\theta \right)}}{c^{2} r}$$

$$R^{0}_{330} = \frac{m \sin^{2}{\left(\theta \right)}}{c^{2} r}$$

$$R^{1}_{001} = \frac{2m \left(c^{2} r - 2m\right)}{c^{2} r^{4}}$$

$$R^{1}_{010} = \frac{2m \left(- c^{2} r + 2m\right)}{c^{2} r^{4}}$$

$$R^{1}_{212} = - \frac{m}{c^{2} r}$$

$$R^{1}_{221} = \frac{m}{c^{2} r}$$

$$R^{1}_{313} = - \frac{m \sin^{2}{\left(\theta \right)}}{c^{2} r}$$

$$R^{1}_{331} = \frac{m \sin^{2}{\left(\theta \right)}}{c^{2} r}$$

$$R^{2}_{002} = \frac{m \left(- c^{2} r + 2m\right)}{c^{2} r^{4}}$$

$$R^{2}_{020} = \frac{m \left(c^{2} r - 2m\right)}{c^{2} r^{4}}$$

$$R^{2}_{112} = \frac{m}{r^{2} \left(c^{2} r - 2m\right)}$$

$$R^{2}_{121} = - \frac{m}{r^{2} \left(c^{2} r - 2m\right)}$$

$$R^{2}_{323} = \frac{2m \sin^{2}{\left(\theta \right)}}{c^{2} r}$$

$$R^{2}_{332} = - \frac{2m \sin^{2}{\left(\theta \right)}}{c^{2} r}$$

$$R^3_{003} = \frac{m\left(-c^2r + 2m\right)}{c^2r^4}$$
$$R^3_{030} = \frac{m\left(c^2r - 2m\right)}{c^2r^4}$$
$$R^3_{113} = \frac{m}{r^2\left(c^2r - 2m\right)}$$
$$R^3_{131} = -\frac{m}{r^2\left(c^2r - 2m\right)}$$
$$R^3_{223} = -\frac{2m}{c^2r}$$
$$R^3_{232} = \frac{2m}{c^2r}$$

4 Perihelion effect on an object revolving around a schwarzschild blackhole

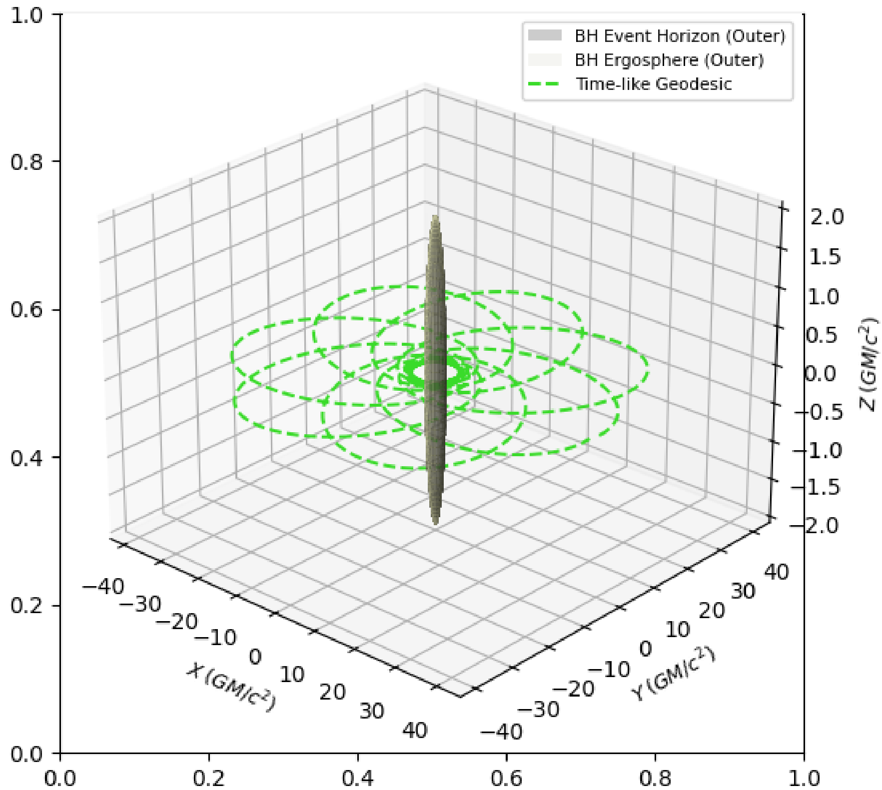

Figure 1:

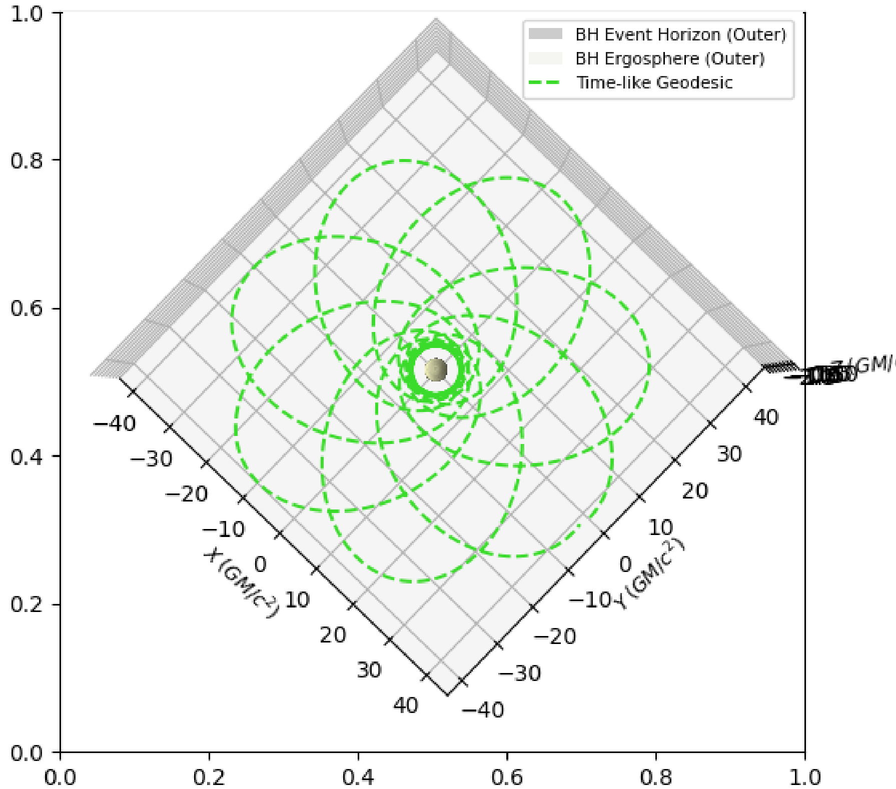

Figure 2:

5 Code for the above computations and images

5.1 For a general diagonal $g_{\mu\nu}$ with rotational symmetry

```
from einsteinpy.symbolic import (
    MetricTensor,
    ChristoffelSymbols,
    RicciTensor,
    RicciScalar,
    RiemannCurvatureTensor
)
from sympy import symbols, Function, simplify, sin, latex
```

```
def compute_and_export_latex(metric, coordinates, filename="
    tensor_output.tex"):
    dx_ds = symbols('v0 v1 v2 v3')

    metric_tensor = MetricTensor(metric, syms=coordinates)
    christoffel = ChristoffelSymbols.from_metric(metric_tensor)
    ricci_tensor = RicciTensor.from_metric(metric_tensor)
    ricci_scalar = RicciScalar.from_riccitensor(ricci_tensor)
    riemann_tensor = RiemannCurvatureTensor.from_metric(
        metric_tensor)

    with open(filename, "w") as f:
        f.write("\\documentclass{article}\n")
        f.write("\\usepackage{amsmath}\n\\usepackage{geometry}\n\\
            geometry{margin=1in}\n")
        f.write("\\begin{document}\n")

        f.write("\\section*{Non-zero Christoffel Symbols}\n\\begin{
            align*}\n")
        for i in range(metric_tensor.dims):
            for j in range(metric_tensor.dims):
                for k in range(metric_tensor.dims):
                    symbol = simplify(christoffel[i, j, k])
                    if not symbol.is_zero:
                        expr = latex(symbol)
                        f.write(f"\\Gamma^{{{i}}}_{{{j}{k}}} &= {
                            expr} \\\\\n")
        f.write("\\end{align*}\n")

        f.write("\\section*{Geodesic Equations}\n\\begin{align*}\n"
            )
        for alpha in range(metric_tensor.dims):
            total = 0
            for beta in range(metric_tensor.dims):
                for gamma in range(metric_tensor.dims):
                    total += christoffel[alpha, beta, gamma] *
                        dx_ds[beta] * dx_ds[gamma]
            simplified = simplify(-total)
            expr = latex(simplified)
            coord_name = latex(coordinates[alpha])
            f.write(f"\\frac{{d^2 {coord_name}}}{{ds^2}} &= {expr}
                \\\\\n")
        f.write("\\end{align*}\n")

        f.write("\\section*{Non-zero Ricci Tensor Components}\n\\
            begin{align*}\n")
        for i in range(metric_tensor.dims):
            for j in range(metric_tensor.dims):
                rt_component = simplify(ricci_tensor[i, j])
                if not rt_component.is_zero:
                    expr = latex(rt_component)
                    f.write(f"R_{{{i}{j}}} &= {expr} \\\\\n")
        f.write("\\end{align*}\n")

        f.write("\\section*{Ricci Scalar}\n")
        scalar_expr = latex(simplify(ricci_scalar.expr))
        f.write(f"\\[ R = {scalar_expr} \\]\n")
```

```
        f.write("\\section*{Non-zero Riemann Tensor Components}\n\\
            begin{align*}\n")
        for rho in range(metric_tensor.dims):
            for sigma in range(metric_tensor.dims):
                for mu in range(metric_tensor.dims):
                    for nu in range(metric_tensor.dims):
                        r_component = simplify(riemann_tensor[rho,
                            sigma, mu, nu])
                        if not r_component.is_zero:
                            expr = latex(r_component)
                            f.write(f"R^{{{rho}}}_{{{sigma}{mu}{nu
                                }}} &= {expr} \\\\\n")
        f.write("\\end{align*}\n")

        f.write("\\end{document}\n")

    print(f"LaTeX output written to: {filename}")


if __name__ == "__main__":
    t, r, theta, phi = symbols('t r theta phi')

    A = Function('A')(r)
    B = Function('B')(r)

    g_metric = [
        [A, 0, 0, 0],
        [0, -B, 0, 0],
        [0, 0, -r**2, 0],
        [0, 0, 0, -r**2 * sin(theta)**2]
    ]

    coordinates = [t, r, theta, phi]
    compute_and_export_latex(g_metric, coordinates)
```

5.2 For a Schwarzschild blackhole

```
from einsteinpy.symbolic import (
    MetricTensor,
    ChristoffelSymbols,
    RicciTensor,
    RicciScalar,
    RiemannCurvatureTensor
)
from sympy import symbols, Function, simplify, sin, latex

def compute_and_export_latex(metric, coordinates, filename="
    tensor_output.tex"):
    dx_ds = symbols('v0 v1 v2 v3')

    metric_tensor = MetricTensor(metric, syms=coordinates)
    christoffel = ChristoffelSymbols.from_metric(metric_tensor)
    ricci_tensor = RicciTensor.from_metric(metric_tensor)
    ricci_scalar = RicciScalar.from_riccitensor(ricci_tensor)
```

```
    riemann_tensor = RiemannCurvatureTensor.from_metric(
        metric_tensor)

    with open(filename, "w") as f:
        f.write("\\documentclass{article}\n")
        f.write("\\usepackage{amsmath}\n\\usepackage{geometry}\n\\
            geometry{margin=1in}\n")
        f.write("\\begin{document}\n")

        f.write("\\section*{Non-zero Christoffel Symbols}\n\\begin{
            align*}\n")
        for i in range(metric_tensor.dims):
            for j in range(metric_tensor.dims):
                for k in range(metric_tensor.dims):
                    symbol = simplify(christoffel[i, j, k])
                    if not symbol.is_zero:
                        expr = latex(symbol)
                        f.write(f"\\Gamma^{{{i}}}_{{{j}{k}}} &= {
                            expr} \\\\\n")
        f.write("\\end{align*}\n")

        f.write("\\section*{Geodesic Equations}\n\\begin{align*}\n"
            )
        for alpha in range(metric_tensor.dims):
            total = 0
            for beta in range(metric_tensor.dims):
                for gamma in range(metric_tensor.dims):
                    total += christoffel[alpha, beta, gamma] *
                        dx_ds[beta] * dx_ds[gamma]
            simplified = simplify(-total)
            expr = latex(simplified)
            coord_name = latex(coordinates[alpha])
            f.write(f"\\frac{{d^2 {coord_name}}}{{ds^2}} &= {expr}
                \\\\\n")
        f.write("\\end{align*}\n")

        f.write("\\section*{Non-zero Ricci Tensor Components}\n\\
            begin{align*}\n")
        for i in range(metric_tensor.dims):
            for j in range(metric_tensor.dims):
                rt_component = simplify(ricci_tensor[i, j])
                if not rt_component.is_zero:
                    expr = latex(rt_component)
                    f.write(f"R_{{{i}{j}}} &= {expr} \\\\\n")
        f.write("\\end{align*}\n")

        f.write("\\section*{Ricci Scalar}\n")
        scalar_expr = latex(simplify(ricci_scalar.expr))
        f.write(f"\\[ R = {scalar_expr} \\]\n")

        f.write("\\section*{Non-zero Riemann Tensor Components}\n\\
            begin{align*}\n")
        for rho in range(metric_tensor.dims):
            for sigma in range(metric_tensor.dims):
                for mu in range(metric_tensor.dims):
                    for nu in range(metric_tensor.dims):
```

```
                        r_component = simplify(riemann_tensor[rho,
                            sigma, mu, nu])
                        if not r_component.is_zero:
                            expr = latex(r_component)
                            f.write(f"R^{{{rho}}}_{{{sigma}{mu}{nu
                                }}} &= {expr} \\\\\n")
        f.write("\\end{align*}\n")

        f.write("\\end{document}\n")

    print(f"LaTeX output written to: {filename}")


if __name__ == "__main__":
    t, r, theta, phi = symbols('t r theta phi')
    m = symbols('m')
    c = symbols('c')

    g_metric = [
        [c**2*(1 - 2*m/(c**2*r)), 0, 0, 0],
        [0, -1/(1 - 2*m/(c**2*r)), 0, 0],
        [0, 0, -r**2, 0],
        [0, 0, 0, -r**2 * sin(theta)**2]
    ]

    coordinates = [t, r, theta, phi]
    compute_and_export_latex(g_metric, coordinates)
```

5.3 For the simulation and creation of the perihelion effect on an object revolving around a Schwarzschild blackhole

```
import numpy as np

from einsteinpy.geodesic import Timelike
from einsteinpy.plotting.geodesic import GeodesicPlotter

position = [40., np.pi / 2, 0.]
momentum = [0., 0., 3.83405]
a = 0.
steps = 5500
delta = 1

geod = Timelike(
    metric="Schwarzschild",
    metric_params=(a,),
    position=position,
    momentum=momentum,
    steps=steps,
    delta=delta,
    return_cartesian=True
)

gpl = GeodesicPlotter()
```

```
gpl.plot(geod)
gpl.show()
```