example

July 4, 2021

1 GRTC-GUI (General Relativity Tensorial Calculations - GUI)

1.1 Schwarzschild Metric in Geometric units, where $G = c = 1$ and $r_s = 2M$

[1]:
```
from GR_tensors import *
from sympy import init_printing, sin, symbols

init_printing()

# Defining the symbols in the coordinate system
t, r, theta, phi = symbols('t, r, theta, phi')

# Defining some extra symbols
r_s = symbols('r_s')

# Defining the diagonal components of the metric tensor
diag_comp = [-(1-r_s/r), (1-r_s/r)**(-1), r**2, r**2*sin(theta)**2]

#Defining the coordinate system
coord_sys = [t, r, theta, phi]
```

2 Metric Tensor

[2]:
```
# Obtaining the metric tensor
mt = MetricTensor(diag_comp, coord_sys)
metric_tensor = mt.get_metrictensor()
metric_tensor
```

[2]:

$$\begin{bmatrix} \frac{-r+r_s}{r} & 0 & 0 & 0 \\ 0 & \frac{r}{r-r_s} & 0 & 0 \\ 0 & 0 & r^2 & 0 \\ 0 & 0 & 0 & r^2 \sin^2(\theta) \end{bmatrix}$$

[3]:
```
# Default type of the metric tensor
mt.get_metrictensor_type()
```

[3]: 'dd'

[4]:
```python
# Varying type 'dd' metric tensor to 'ud'
mt.vary_metrictensor_type(metric_tensor, 'ud')
```

[4]:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

[5]:
```python
mt.get_metrictensor_type()
```

[5]:
```
'ud'
```

[6]:
```python
# Varying type 'dd' metric tensor to 'uu'
mt.vary_metrictensor_type(metric_tensor, 'uu')
```

[6]:

$$\begin{bmatrix} -\frac{r}{r-r_s} & 0 & 0 & 0 \\ 0 & \frac{r-r_s}{r} & 0 & 0 \\ 0 & 0 & \frac{1}{r^2} & 0 \\ 0 & 0 & 0 & \frac{1}{r^2 \sin^2(\theta)} \end{bmatrix}$$

[7]:
```python
mt.get_metrictensor_type()
```

[7]:
```
'uu'
```

[8]:
```python
# Obtaining the inverse of the metric tensor directly
mt.get_inverse()
```

[8]:

$$\begin{bmatrix} -\frac{r}{r-r_s} & 0 & 0 & 0 \\ 0 & \frac{r-r_s}{r} & 0 & 0 \\ 0 & 0 & \frac{1}{r^2} & 0 \\ 0 & 0 & 0 & \frac{1}{r^2 \sin^2(\theta)} \end{bmatrix}$$

3 Christoffel Symbol

[9]:
```python
# Obtaining the Christoffel Symbol
cs = ChristoffelSymbol(diag_comp, coord_sys)
chris_symbol = cs.get_christoffelsymbol()
chris_symbol
```

[9]:

$$\left[\begin{bmatrix} 0 & \frac{r_s}{2r(r-r_s)} & 0 & 0 \\ \frac{r_s}{2r(r-r_s)} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{r_s(r-r_s)}{2r^3} & 0 & 0 & 0 \\ 0 & -\frac{r_s}{2r(r-r_s)} & 0 & 0 \\ 0 & 0 & -r+r_s & 0 \\ 0 & 0 & 0 & (-r+r_s)\sin^2(\theta) \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{r} & 0 \\ 0 & \frac{1}{r} & 0 & 0 \\ 0 & 0 & 0 & -\frac{\sin(2\theta)}{2} \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & \frac{1}{r} \end{bmatrix} \right.$$

[10]:
```python
# Default type of the Christoffel Symbol
cs.get_christoffelsymbol_type()
```

[10]:
```
'udd'
```

[11]:
```python
# Varying type 'udd' Christoffel Symbol to 'ddd'
chris_symbol03 = cs.vary_christoffelsymbol_type(chris_symbol, 'ddd')
chris_symbol03
```

[11]:

$$\left[\begin{bmatrix} 0 & -\frac{r_s}{2r^2} & 0 & 0 \\ -\frac{r_s}{2r^2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{r_s}{2r^2} & 0 & 0 & 0 \\ 0 & -\frac{r_s}{2(r-r_s)^2} & 0 & 0 \\ 0 & 0 & -r & 0 \\ 0 & 0 & 0 & -r\sin^2(\theta) \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & r & 0 \\ 0 & r & 0 & 0 \\ 0 & 0 & 0 & -\frac{r^2\sin(2\theta)}{2} \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & r\sin^2(\theta) & \frac{r^2\sin(2\theta)}{2} \end{bmatrix}$$

[12]:
```python
cs.get_christoffelsymbol_type()
```

[12]: 'ddd'

[13]:
```python
# Obtaining the non-zero components of the given Christoffel Symbol for type ddd
cs.nonzero_christoffelsymbol(chris_symbol03)
```

$\Gamma_{ttr} = -\frac{r_s}{2r^2}$

$\Gamma_{trt} = -\frac{r_s}{2r^2}$

$\Gamma_{rtt} = \frac{r_s}{2r^2}$

$\Gamma_{rrr} = -\frac{r_s}{2(r-r_s)^2}$

$\Gamma_{r\theta\theta} = -r$

$\Gamma_{r\phi\phi} = -r\sin^2(\theta)$

$\Gamma_{\theta r\theta} = r$

$\Gamma_{\theta\theta r} = r$

$\Gamma_{\theta\phi\phi} = -\frac{r^2\sin(2\theta)}{2}$

$\Gamma_{\phi r\phi} = r\sin^2(\theta)$

$\Gamma_{\phi\theta\phi} = \frac{r^2\sin(2\theta)}{2}$

$\Gamma_{\phi\phi r} = r\sin^2(\theta)$

$\Gamma_{\phi\phi\theta} = \frac{r^2\sin(2\theta)}{2}$

[14]:
```python
# Varying type 'udd' Christoffel Symbol to 'uud'
chris_symbol21 = cs.vary_christoffelsymbol_type(chris_symbol, 'uud')
chris_symbol21
```

[14]:

$$\left[\begin{bmatrix} 0 & -\frac{r_s}{2(r-r_s)^2} & 0 & 0 \\ \frac{r_s}{2r^2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} -\frac{r_s}{2r^2} & 0 & 0 & 0 \\ 0 & -\frac{r_s}{2r^2} & 0 & 0 \\ 0 & 0 & \frac{-r+r_s}{r^2} & 0 \\ 0 & 0 & 0 & \frac{-r+r_s}{r^2} \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{r-r_s}{r^2} & 0 \\ 0 & \frac{1}{r^3} & 0 & 0 \\ 0 & 0 & 0 & -\frac{1}{r^2\tan(\theta)} \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \frac{1}{r^3\sin^2(\theta)} & \frac{\cos(\theta)}{r^2\sin^3(\theta)} \end{bmatrix}$$

[15]:
```python
cs.get_christoffelsymbol_type()
```

[15]: 'uud'

[16]:
```python
# Varying type 'udd' Christoffel Symbol to 'uuu'
chris_symbol30 = cs.vary_christoffelsymbol_type(chris_symbol, 'uuu')
chris_symbol30
```

[16]:

$$\left[\begin{bmatrix} 0 & -\frac{r_s}{2r(r-r_s)} & 0 & 0 \\ -\frac{r_s}{2r(r-r_s)} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{r_s}{2r(r-r_s)} & 0 & 0 & 0 \\ 0 & \frac{r_s(-r+r_s)}{2r^3} & 0 & 0 \\ 0 & 0 & \frac{-r+r_s}{r^4} & 0 \\ 0 & 0 & 0 & \frac{-r+r_s}{r^4\sin^2(\theta)} \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{r-r_s}{r^4} & 0 \\ 0 & \frac{r-r_s}{r^4} & 0 & 0 \\ 0 & 0 & 0 & -\frac{\cos(\theta)}{r^4\sin^3(\theta)} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}\right.$$

[17]:
```python
cs.get_christoffelsymbol_type()
```

[17]: 'uuu'

[18]:
```python
cs.nonzero_christoffelsymbol(chris_symbol30)
```

$$\Gamma^{ttr} = -\frac{r_s}{2r(r-r_s)}$$

$$\Gamma^{trt} = -\frac{r_s}{2r(r-r_s)}$$

$$\Gamma^{rtt} = \frac{r_s}{2r(r-r_s)}$$

$$\Gamma^{rrr} = \frac{r_s(-r+r_s)}{2r^3}$$

$$\Gamma^{r\theta\theta} = \frac{-r+r_s}{r^4}$$

$$\Gamma^{r\phi\phi} = \frac{-r+r_s}{r^4\sin^2(\theta)}$$

$$\Gamma^{\theta r\theta} = \frac{r-r_s}{r^4}$$

$$\Gamma^{\theta\theta r} = \frac{r-r_s}{r^4}$$

$$\Gamma^{\theta\phi\phi} = -\frac{\cos(\theta)}{r^4\sin^3(\theta)}$$

$$\Gamma^{\phi r\phi} = \frac{r-r_s}{r^4\sin^2(\theta)}$$

$$\Gamma^{\phi\theta\phi} = \frac{\cos(\theta)}{r^4\sin^3(\theta)}$$

$$\Gamma^{\phi\phi r} = \frac{r-r_s}{r^4\sin^2(\theta)}$$

$$\Gamma^{\phi\phi\theta} = \frac{\cos(\theta)}{r^4\sin^3(\theta)}$$

4 Riemann Tensor

[19]:
```python
# Obtaining the Riemann Tensor
rt = RiemannTensor(diag_comp, coord_sys)
riemann_tensor = rt.get_riemanntensor()
riemann_tensor
```

[19]:

$$\left[\begin{matrix}\left[\begin{matrix}0 & 0 & 0 & 0\\0 & - \frac{r_{s}}{r^{2} \left(r - r_{s}\right)} & 0 & 0\\0 & 0 & \frac{r_{s}}{2 r} & 0\\0 & 0 & 0 & \frac{r_{s} \sin^{2}{\left(\theta \right)}}{2 r}\end{matrix}\right] & \left[\begin{matrix}0 & \frac{r_{s}}{r^{2} \left(r - r_{s}\right)} & 0 & 0\\0 & 0 & 0 & 0\\0 & 0 & 0 & 0\\0 & 0 & 0 & 0\end{matrix}\right] & \left[\begin{matrix}0 & 0 & - \frac{r_{s}}{2 r} & 0\\0 & 0 & 0 & 0\\0 & 0 & 0 & 0\\0 & 0 & 0 & 0\end{matrix}\right]\\\left[\begin{matrix}0 & 0 & 0 & 0\\\frac{r_{s} \left(- r + r_{s}\right)}{r^{4}} & 0 & 0 & 0\\0 & 0 & 0 & 0\\0 & 0 & 0 & 0\end{matrix}\right] & \left[\begin{matrix}\frac{r_{s} \left(r - r_{s}\right)}{r^{4}} & 0 & 0 & 0\\0 & 0 & 0 & 0\\0 & 0 & \frac{r_{s}}{2 r} & 0\\0 & 0 & 0 & \frac{r_{s} \sin^{2}{\left(\theta \right)}}{2 r}\end{matrix}\right] & \left[\begin{matrix}0 & 0 & 0 & 0\\0 & 0 & - \frac{r_{s}}{2 r} & 0\\0 & 0 & 0 & 0\\0 & 0 & 0 & 0\end{matrix}\right]\\\left[\begin{matrix}0 & 0 & 0 & 0\\0 & 0 & 0 & 0\\\frac{r_{s} \left(r - r_{s}\right)}{2 r^{4}} & 0 & 0 & 0\\0 & 0 & 0 & 0\end{matrix}\right] & \left[\begin{matrix}0 & 0 & 0 & 0\\0 & 0 & 0 & 0\\0 & - \frac{r_{s}}{2 r^{2} \left(r - r_{s}\right)} & 0 & 0\\0 & 0 & 0 & 0\end{matrix}\right] & \left[\begin{matrix}\frac{r_{s} \left(- r + r_{s}\right)}{2 r^{4}} & 0 & 0 & 0\\0 & \frac{r_{s}}{2 r^{2} \left(r - r_{s}\right)} & 0 & 0\\0 & 0 & 0 & 0\\0 & 0 & 0 & - \frac{r_{s} \sin^{2}{\left(\theta \right)}}{r}\end{matrix}\right]\\\left[\begin{matrix}0 & 0 & 0 & 0\\0 & 0 & 0 & 0\\0 & 0 & 0 & 0\\\frac{r_{s} \left(r - r_{s}\right)}{2 r^{4}} & 0 & 0 & 0\end{matrix}\right] & \left[\begin{matrix}0 & 0 & 0 & 0\\0 & 0 & 0 & 0\\0 & 0 & 0 & 0\\0 & - \frac{r_{s}}{2 r^{2} \left(r - r_{s}\right)} & 0 & 0\end{matrix}\right] & \left[\begin{matrix}0 & 0 & 0 & 0\\0 & 0 & 0 & 0\\0 & 0 & 0 & 0\\0 & 0 & \frac{r_{s}}{r} & 0\end{matrix}\right]\end{matrix}\right.$$

[20]:
```python
# Default type of the Riemann Tensor
rt.get_riemanntensor_type()
```

[20]: 'uddd'

[21]:
```python
# Varying type 'uddd' Riemann Tensor to 'dddd'
riemann_tensor04 = rt.vary_riemanntensor_type(riemann_tensor, 'dddd')
riemann_tensor04
```

[21]:

$$\left[\begin{matrix}\left[\begin{matrix}0 & 0 & 0 & 0\\0 & \frac{r_{s}}{r^{3}} & 0 & 0\\0 & 0 & \frac{r_{s} \left(- r + r_{s}\right)}{2 r^{2}} & 0\\0 & 0 & 0 & \frac{r_{s} \left(- r + r_{s}\right) \sin^{2}{\left(\theta \right)}}{2 r^{2}}\end{matrix}\right] & \left[\begin{matrix}0 & - \frac{r_{s}}{r^{3}} & 0 & 0\\0 & 0 & 0 & 0\\0 & 0 & 0 & 0\\0 & 0 & 0 & 0\end{matrix}\right] & \left[\begin{matrix}0 & 0 & \frac{r_{s} \left(r - r_{s}\right)}{2 r^{2}} & 0\\0 & 0 & 0 & 0\\0 & 0 & 0 & 0\\0 & 0 & 0 & 0\end{matrix}\right]\\\left[\begin{matrix}0 & 0 & 0 & 0\\- \frac{r_{s}}{r^{3}} & 0 & 0 & 0\\0 & 0 & 0 & 0\\0 & 0 & 0 & 0\end{matrix}\right] & \left[\begin{matrix}\frac{r_{s}}{r^{3}} & 0 & 0 & 0\\0 & 0 & 0 & 0\\0 & 0 & \frac{r_{s}}{2 \left(r - r_{s}\right)} & 0\\0 & 0 & 0 & \frac{r_{s} \sin^{2}{\left(\theta \right)}}{2 \left(r - r_{s}\right)}\end{matrix}\right] & \left[\begin{matrix}0 & 0 & 0 & 0\\0 & 0 & - \frac{r_{s}}{2 r - 2 r_{s}} & 0\\0 & 0 & 0 & 0\\0 & 0 & 0 & 0\end{matrix}\right]\\\left[\begin{matrix}0 & 0 & 0 & 0\\0 & 0 & 0 & 0\\\frac{r_{s} \left(r - r_{s}\right)}{2 r^{2}} & 0 & 0 & 0\\0 & 0 & 0 & 0\end{matrix}\right] & \left[\begin{matrix}0 & 0 & 0 & 0\\0 & 0 & 0 & 0\\0 & - \frac{r_{s}}{2 r - 2 r_{s}} & 0 & 0\\0 & 0 & 0 & 0\end{matrix}\right] & \left[\begin{matrix}\frac{r_{s} \left(- r + r_{s}\right)}{2 r^{2}} & 0 & 0 & 0\\0 & \frac{r_{s}}{2 \left(r - r_{s}\right)} & 0 & 0\\0 & 0 & 0 & 0\\0 & 0 & 0 & - r r_{s} \sin^{2}{\left(\theta \right)}\end{matrix}\right]\\\left[\begin{matrix}0 & 0 & 0 & 0\\0 & 0 & 0 & 0\\0 & 0 & 0 & 0\\\frac{r_{s} \left(r - r_{s}\right) \sin^{2}{\left(\theta \right)}}{2 r^{2}} & 0 & 0 & 0\end{matrix}\right] & \left[\begin{matrix}0 & 0 & 0 & 0\\0 & 0 & 0 & 0\\0 & 0 & 0 & 0\\0 & - \frac{r_{s} \sin^{2}{\left(\theta \right)}}{2 r - 2 r_{s}} & 0 & 0\end{matrix}\right] & \left[\begin{matrix}0 & 0 & 0 & 0\\0 & 0 & 0 & 0\\0 & 0 & 0 & 0\\0 & 0 & r r_{s} \sin^{2}{\left(\theta \right)} & 0\end{matrix}\right]\end{matrix}\right.$$

[22]:
```python
rt.get_riemanntensor_type()
```

[22]: 'dddd'

[23]:
```
rt.nonzero_riemanntensor(riemann_tensor04)
```

$R_{ttrr} = \frac{r_s}{r^3}$

$R_{tt\theta\theta} = \frac{r_s(-r+r_s)}{2r^2}$

$R_{tt\phi\phi} = \frac{r_s(-r+r_s)\sin^2(\theta)}{2r^2}$

$R_{trtr} = -\frac{r_s}{r^3}$

$R_{t\theta t\theta} = \frac{r_s(r-r_s)}{2r^2}$

$R_{t\phi t\phi} = \frac{r_s(r-r_s)\sin^2(\theta)}{2r^2}$

$R_{rtrt} = -\frac{r_s}{r^3}$

$R_{rrtt} = \frac{r_s}{r^3}$

$R_{rr\theta\theta} = \frac{r_s}{2(r-r_s)}$

$R_{rr\phi\phi} = \frac{r_s\sin^2(\theta)}{2(r-r_s)}$

$R_{r\theta r\theta} = -\frac{r_s}{2r-2r_s}$

$R_{r\phi r\phi} = -\frac{r_s\sin^2(\theta)}{2r-2r_s}$

$R_{\theta t\theta t} = \frac{r_s(r-r_s)}{2r^2}$

$R_{\theta r\theta r} = -\frac{r_s}{2r-2r_s}$

$R_{\theta\theta tt} = \frac{r_s(-r+r_s)}{2r^2}$

$R_{\theta\theta rr} = \frac{r_s}{2(r-r_s)}$

$R_{\theta\theta\phi\phi} = -rr_s\sin^2(\theta)$

$R_{\theta\phi\theta\phi} = rr_s\sin^2(\theta)$

$R_{\phi t\phi t} = \frac{r_s(r-r_s)\sin^2(\theta)}{2r^2}$

$R_{\phi r\phi r} = -\frac{r_s\sin^2(\theta)}{2r-2r_s}$

$R_{\phi\theta\phi\theta} = rr_s\sin^2(\theta)$

$R_{\phi\phi tt} = \frac{r_s(-r+r_s)\sin^2(\theta)}{2r^2}$

$R_{\phi\phi rr} = \frac{r_s\sin^2(\theta)}{2(r-r_s)}$

$R_{\phi\phi\theta\theta} = -rr_s\sin^2(\theta)$

5 Ricci Tensor

[24]:
```
# Obtaining the Ricci Tensor
rit = RicciTensor(diag_comp, coord_sys)
ricci_tensor = rit.get_riccitensor()
ricci_tensor
```

[24]:

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

[25]:
```
# Default type of the Ricci Tensor
rit.get_riccitensor_type()
```

[25]: 'dd'

[26]:
```
# Varying type 'dd' Ricci Tensor to 'uu'
rit.vary_riccitensor_type(ricci_tensor, 'uu')
```

[26]:

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

[27]:
```
rit.get_riccitensor_type()
```

[27]: 'uu'

6 Ricci Scalar

[28]:
```
# Obtaining the Ricci Scalar
rs = RicciScalar(diag_comp, coord_sys)
ricci_scalar = rs.get_ricciscalar()
ricci_scalar
```

[28]:

$$0$$

7 Traceless Ricci Tensor

[29]:
```
# Obtaining the Traceless Ricci Tensor
trt = TracelessRicciTensor(diag_comp, coord_sys)
traceless_ricci_tensor = trt.get_trclss_riccitensor()
traceless_ricci_tensor
```

[29]:

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

[30]:
```python
# Default type of the Traceless Ricci Tensor
trt.get_trclss_riccitensor_type()
```

[30]: 'dd'

[31]:
```python
# Varying type 'dd' Traceless Ricci Tensor to 'uu'
trt.vary_trclss_riccitensor_type(traceless_ricci_tensor, 'uu')
```

[31]:
$$\begin{bmatrix}0 & 0 & 0 & 0\\0 & 0 & 0 & 0\\0 & 0 & 0 & 0\\0 & 0 & 0 & 0\end{bmatrix}$$

[32]:
```python
trt.get_trclss_riccitensor_type()
```

[32]: 'uu'

8 Weyl Tensor

[33]:
```python
# Obtaining the Weyl Tensor
wyl = WeylTensor(diag_comp, coord_sys)
weyl_tensor = wyl.get_weyltensor()
weyl_tensor
```

[33]:
$$\left[\begin{matrix}
\begin{bmatrix}0 & 0 & 0 & 0\\0 & \frac{r_s}{r^3} & 0 & 0\\0 & 0 & \frac{r_s(-r+r_s)}{2r^2} & 0\\0 & 0 & 0 & \frac{r_s(-r+r_s)\sin^2(\theta)}{2r^2}\end{bmatrix} & \begin{bmatrix}0 & -\frac{r_s}{r^3} & 0 & 0\\0 & 0 & 0 & 0\\0 & 0 & 0 & 0\\0 & 0 & 0 & 0\end{bmatrix} & \begin{bmatrix}0 & 0 & \frac{r_s(r-r_s)}{2r^2} & 0\\0 & 0 & 0 & 0\\0 & 0 & 0 & 0\\0 & 0 & 0 & 0\end{bmatrix} & \cdots\\
\begin{bmatrix}0 & 0 & 0 & 0\\-\frac{r_s}{r^3} & 0 & 0 & 0\\0 & 0 & 0 & 0\\0 & 0 & 0 & 0\end{bmatrix} & \begin{bmatrix}\frac{r_s}{r^3} & 0 & 0 & 0\\0 & 0 & 0 & 0\\0 & 0 & \frac{r_s}{2(r-r_s)} & 0\\0 & 0 & 0 & \frac{r_s\sin^2(\theta)}{2(r-r_s)}\end{bmatrix} & \begin{bmatrix}0 & 0 & 0 & 0\\0 & 0 & -\frac{r_s}{2r-2r_s} & 0\\0 & 0 & 0 & 0\\0 & 0 & 0 & 0\end{bmatrix} & \cdots\\
\begin{bmatrix}0 & 0 & 0 & 0\\0 & 0 & 0 & 0\\\frac{r_s(r-r_s)}{2r^2} & 0 & 0 & 0\\0 & 0 & 0 & 0\end{bmatrix} & \begin{bmatrix}0 & 0 & 0 & 0\\0 & 0 & 0 & 0\\0 & -\frac{r_s}{2r-2r_s} & 0 & 0\\0 & 0 & 0 & 0\end{bmatrix} & \begin{bmatrix}\frac{r_s(-r+r_s)}{2r^2} & 0 & 0 & 0\\0 & \frac{r_s}{2(r-r_s)} & 0 & 0\\0 & 0 & 0 & 0\\0 & 0 & 0 & -rr_s\sin^2(\theta)\end{bmatrix} & \cdots\\
\begin{bmatrix}0 & 0 & 0 & 0\\0 & 0 & 0 & 0\\0 & 0 & 0 & 0\\\frac{r_s(r-r_s)\sin^2(\theta)}{2r^2} & 0 & 0 & 0\end{bmatrix} & \begin{bmatrix}0 & 0 & 0 & 0\\0 & 0 & 0 & 0\\0 & 0 & 0 & 0\\0 & -\frac{r_s\sin^2(\theta)}{2r-2r_s} & 0 & 0\end{bmatrix} & \begin{bmatrix}0 & 0 & 0 & 0\\0 & 0 & 0 & 0\\0 & 0 & 0 & 0\\0 & 0 & rr_s\sin^2(\theta) & 0\end{bmatrix} & \cdots
\end{matrix}\right.$$

[34]:
```python
# Default type of the Weyl Tensor
wyl.get_weyltensor_type()
```

[34]: 'dddd'

[35]:
```python
# Varying type 'dddd' Weyl Tensor to 'uuuu'
weyl_tensor40 = wyl.vary_weyltensor_type(weyl_tensor, 'uuuu')
weyl_tensor40
```

[35]:

$$\left[\begin{matrix}
\begin{bmatrix}0 & 0 & 0 & 0\\ 0 & \frac{r_s}{r^3} & 0 & 0\\ 0 & 0 & -\frac{r_s}{2r^4(r-r_s)} & 0\\ 0 & 0 & 0 & -\frac{r_s}{2r^4(r-r_s)\sin^2(\theta)}\end{bmatrix} & \begin{bmatrix}0 & -\frac{r_s}{r^3} & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\end{bmatrix} & \begin{bmatrix}0 & 0 & \frac{r_s}{2r^4(r-r_s)} & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\end{bmatrix}\\
\begin{bmatrix}0 & 0 & 0 & 0\\ -\frac{r_s}{r^3} & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\end{bmatrix} & \begin{bmatrix}\frac{r_s}{r^3} & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & \frac{r_s(r-r_s)}{2r^6} & 0\\ 0 & 0 & 0 & \frac{r_s(r-r_s)}{2r^6\sin^2(\theta)}\end{bmatrix} & \begin{bmatrix}0 & 0 & 0 & 0\\ 0 & 0 & \frac{r_s(-r+r_s)}{2r^6} & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\end{bmatrix}\\
\begin{bmatrix}0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ \frac{r_s}{2r^4(r-r_s)} & 0 & 0 & 0\\ 0 & 0 & 0 & 0\end{bmatrix} & \begin{bmatrix}0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & \frac{r_s(-r+r_s)}{2r^6} & 0 & 0\\ 0 & 0 & 0 & 0\end{bmatrix} & \begin{bmatrix}-\frac{r_s}{2r^4(r-r_s)} & 0 & 0 & 0\\ 0 & \frac{r_s(r-r_s)}{2r^6} & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & -\frac{r_s}{r^7\sin^2(\theta)}\end{bmatrix}\\
\begin{bmatrix}0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ \frac{r_s}{2r^4(r-r_s)\sin^2(\theta)} & 0 & 0 & 0\end{bmatrix} & \begin{bmatrix}0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & \frac{r_s(-r+r_s)}{2r^6\sin^2(\theta)} & 0 & 0\end{bmatrix} & \begin{bmatrix}0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & \frac{r_s}{r^7\sin^2(\theta)} & 0\end{bmatrix}
\end{matrix}\right.$$

[36]:
```python
wyl.get_weyltensor_type()
```

[36]: 'uuuu'

[37]:
```python
wyl.nonzero_weyltensor(weyl_tensor40)
```

$$C^{ttrr} = \frac{r_s}{r^3}$$

$$C^{tt\theta\theta} = -\frac{r_s}{2r^4(r-r_s)}$$

$$C^{tt\phi\phi} = -\frac{r_s}{2r^4(r-r_s)\sin^2(\theta)}$$

$$C^{trtr} = -\frac{r_s}{r^3}$$

$$C^{t\theta t\theta} = \frac{r_s}{2r^4(r-r_s)}$$

$$C^{t\phi t\phi} = \frac{r_s}{2r^4(r-r_s)\sin^2(\theta)}$$

$$C^{rtrt} = -\frac{r_s}{r^3}$$

$$C^{rrtt} = \frac{r_s}{r^3}$$

$$C^{rr\theta\theta} = \frac{r_s(r-r_s)}{2r^6}$$

$$C^{rr\phi\phi} = \frac{r_s(r-r_s)}{2r^6\sin^2(\theta)}$$

$$C^{r\theta r\theta} = \frac{r_s(-r+r_s)}{2r^6}$$

$$C^{r\phi r\phi} = \frac{r_s(-r+r_s)}{2r^6\sin^2(\theta)}$$

$C^{\theta t \theta t} = \frac{r_s}{2r^4(r-r_s)}$

$C^{\theta r \theta r} = \frac{r_s(-r+r_s)}{2r^6}$

$C^{\theta \theta tt} = -\frac{r_s}{2r^4(r-r_s)}$

$C^{\theta \theta rr} = \frac{r_s(r-r_s)}{2r^6}$

$C^{\theta \theta \phi \phi} = -\frac{r_s}{r^7 \sin^2(\theta)}$

$C^{\theta \phi \theta \phi} = \frac{r_s}{r^7 \sin^2(\theta)}$

$C^{\phi t \phi t} = \frac{r_s}{2r^4(r-r_s)\sin^2(\theta)}$

$C^{\phi r \phi r} = \frac{r_s(-r+r_s)}{2r^6 \sin^2(\theta)}$

$C^{\phi \theta \phi \theta} = \frac{r_s}{r^7 \sin^2(\theta)}$

$C^{\phi \phi tt} = -\frac{r_s}{2r^4(r-r_s)\sin^2(\theta)}$

$C^{\phi \phi rr} = \frac{r_s(r-r_s)}{2r^6 \sin^2(\theta)}$

$C^{\phi \phi \theta \theta} = -\frac{r_s}{r^7 \sin^2(\theta)}$

9 Einstein Tensor

[38]:
```
# Obtaining the Einstein Tensor
et = EinsteinTensor(diag_comp, coord_sys)
einstein_tensor = et.get_einsteintensor()
einstein_tensor
```

[38]:

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

[39]:
```
# Default type of the Einstein Tensor
et.get_einsteintensor_type()
```

[39]: 'dd'

[40]:
```
# Varying type 'dd' Einstein Tensor to 'uu'
et.vary_einsteintensor_type(einstein_tensor, 'uu')
```

[40]:

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

[41]:
```
et.get_einsteintensor_type()
```

[41]: 'uu'

10 Kretschmann Scalar

[42]:
```
ks = KretschmannScalar(diag_comp, coord_sys)
kret_scalar = ks.get_kretschmannscalar()
kret_scalar
```

[42]: $\frac{12r_s^2}{r^6}$