example

May 11, 2021

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

init_printing()

# Coordinate system that we will work on
coord_sys = symbols('t r theta phi')

# Defining some extra symbols
G, m, a = symbols('G, m, a')

# Defining the diagonal components of the metric tensor
diag_comp = [-1, a**(-1), a**2, a * sin(coord_sys[2])**2]
```

1 Metric Tensor

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

[2]:
$$\begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & \frac{1}{a} & 0 & 0 \\ 0 & 0 & a^2 & 0 \\ 0 & 0 & 0 & a\sin^2(\theta) \end{bmatrix}$$

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

[3]:
```
'dd'
```

[4]:
```
# 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]:
```
mt.get_metrictensor_type()
```

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

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

[6]:
$$\begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & a & 0 & 0 \\ 0 & 0 & \frac{1}{a^2} & 0 \\ 0 & 0 & 0 & \frac{1}{a \sin^2(\theta)} \end{bmatrix}$$

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

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

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

[8]:
$$\begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & a & 0 & 0 \\ 0 & 0 & \frac{1}{a^2} & 0 \\ 0 & 0 & 0 & \frac{1}{a \sin^2(\theta)} \end{bmatrix}$$

2 Christoffel Symbol

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

[9]:
$$\begin{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{\sin(\theta)\cos(\theta)}{a} \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\cos(\theta)}{\sin(\theta)} \\ 0 & 0 & \frac{\cos(\theta)}{\sin(\theta)} & 0 \end{bmatrix} \end{bmatrix}$$

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

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

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

[11]:
$$\begin{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -a\sin(\theta)\cos(\theta) \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & a\sin(\theta)\cos(\theta) \\ 0 & 0 & a\sin(\theta)\cos(\theta) & 0 \end{bmatrix} \end{bmatrix}$$

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

[12]:
```
'ddd'
```

[13]:
```
# Varying type 'udd' Christoffel Symbol to 'uud'
cs.vary_christoffelsymbol_type(chris_symbol, 'uud')
```

[13]:
$$\left[\begin{bmatrix}0 & 0 & 0 & 0\\0 & 0 & 0 & 0\\0 & 0 & 0 & 0\\0 & 0 & 0 & 0\end{bmatrix} \quad \begin{bmatrix}0 & 0 & 0 & 0\\0 & 0 & 0 & 0\\0 & 0 & 0 & 0\\0 & 0 & 0 & 0\end{bmatrix} \quad \begin{bmatrix}0 & 0 & 0 & 0\\0 & 0 & 0 & 0\\0 & 0 & 0 & 0\\0 & 0 & 0 & -\frac{\cos(\theta)}{a^2\sin(\theta)}\end{bmatrix} \quad \begin{bmatrix}0 & 0 & 0 & 0\\0 & 0 & 0 & 0\\0 & 0 & 0 & \frac{\cos(\theta)}{a^2\sin(\theta)}\\0 & 0 & \frac{\cos(\theta)}{a\sin^3(\theta)} & 0\end{bmatrix}\right]$$

[14]:
```
cs.get_christoffelsymbol_type()
```

[14]:
```
'uud'
```

[15]:
```
# Varying type 'udd' Christoffel Symbol to 'uuu'
cs.vary_christoffelsymbol_type(chris_symbol, 'uuu')
```

[15]:
$$\left[\begin{bmatrix}0 & 0 & 0 & 0\\0 & 0 & 0 & 0\\0 & 0 & 0 & 0\\0 & 0 & 0 & 0\end{bmatrix} \quad \begin{bmatrix}0 & 0 & 0 & 0\\0 & 0 & 0 & 0\\0 & 0 & 0 & 0\\0 & 0 & 0 & 0\end{bmatrix} \quad \begin{bmatrix}0 & 0 & 0 & 0\\0 & 0 & 0 & 0\\0 & 0 & 0 & 0\\0 & 0 & 0 & -\frac{\cos(\theta)}{a^3\sin^3(\theta)}\end{bmatrix} \quad \begin{bmatrix}0 & 0 & 0 & 0\\0 & 0 & 0 & 0\\0 & 0 & 0 & \frac{\cos(\theta)}{a^3\sin^3(\theta)}\\0 & 0 & \frac{\cos(\theta)}{a^3\sin^3(\theta)} & 0\end{bmatrix}\right]$$

[16]:
```
cs.get_christoffelsymbol_type()
```

[16]:
```
'uuu'
```

[17]:
```
# Obtaining the non-zero components of the given Christoffel Symbol
cs.nonzero_christoffelsymbol(chris_symbol)
```

$$\Gamma^{\theta}{}_{\phi\phi} = -\frac{\sin(\theta)\cos(\theta)}{a}$$

$$\Gamma^{\phi}{}_{\theta\phi} = \frac{\cos(\theta)}{\sin(\theta)}$$

$$\Gamma^{\phi}{}_{\phi\theta} = \frac{\cos(\theta)}{\sin(\theta)}$$

3 Riemann Tensor

[18]:
```
# Obtaining the Riemann Tensor
rt = RiemannTensor(diag_comp, coord_sys)
riemann_tensor = rt.get_riemanntensor()
riemann_tensor
```

[18]:

$$\left[\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{bmatrix}\quad\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{bmatrix}\quad\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{bmatrix}\quad\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{bmatrix}\right.$$
$$\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{bmatrix}\quad\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{bmatrix}\quad\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{bmatrix}\quad\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{bmatrix}$$
$$\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{bmatrix}\quad\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{bmatrix}\quad\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&-\frac{\sin^2(\theta)}{a}\end{bmatrix}\quad\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&0&\frac{\sin^2(\theta)}{a}\\0&0&0&0\end{bmatrix}$$
$$\left.\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{bmatrix}\quad\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{bmatrix}\quad\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&1&0\end{bmatrix}\quad\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&-1&0\\0&0&0&0\end{bmatrix}\right]$$

[19]:
```
# Default type of the Riemann Tensor
rt.get_riemanntensor_type()
```

[19]: 'uddd'

[20]:
```
# Varying type 'uddd' Riemann Tensor to 'dddd'
rt.vary_riemanntensor_type(riemann_tensor, 'dddd')
```

[20]:
$$\left[\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{bmatrix}\quad\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{bmatrix}\quad\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{bmatrix}\quad\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{bmatrix}\right.$$
$$\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{bmatrix}\quad\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{bmatrix}\quad\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{bmatrix}\quad\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{bmatrix}$$
$$\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{bmatrix}\quad\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{bmatrix}\quad\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&-a\sin^2(\theta)\end{bmatrix}\quad\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&0&a\sin^2(\theta)\\0&0&0&0\end{bmatrix}$$
$$\left.\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{bmatrix}\quad\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{bmatrix}\quad\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&0&a\sin^2(\theta)\\0&0&0&0\end{bmatrix}\quad\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&-a\sin^2(\theta)&0\\0&0&0&0\end{bmatrix}\right]$$

[21]:
```
rt.get_riemanntensor_type()
```

[21]: 'dddd'

4 Ricci Tensor

[22]:
```
# Obtaining the Ricci Tensor
rit = RicciTensor(diag_comp, coord_sys)
ricci_tensor = rit.get_riccitensor()
ricci_tensor
```

[22]:
$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \frac{\sin^2(\theta)}{a} \end{bmatrix}$$

[23]:
```
# Default type of the Ricci Tensor
rit.get_riccitensor_type()
```

[23]:
```
'dd'
```

[24]:
```
# Varying type 'dd' Ricci Tensor to 'uu'
rit.vary_riccitensor_type(ricci_tensor, 'uu')
```

[24]:
$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{a^4} & 0 \\ 0 & 0 & 0 & \frac{1}{a^3 \sin^2(\theta)} \end{bmatrix}$$

[25]:
```
rit.get_riccitensor_type()
```

[25]:
```
'uu'
```

5 Ricci Scalar

[26]:
```
# Obtaining the Ricci Scalar
rs = RicciScalar(diag_comp, coord_sys)
ricci_scalar = rs.get_ricciscalar()
ricci_scalar
```

[26]:
$$\frac{2}{a^2}$$

6 Traceless Ricci Tensor

[27]:
```
# Obtaining the Traceless Ricci Tensor
trt = TracelessRicciTensor(diag_comp, coord_sys)
tracless_ricci_tensor = trt.get_trclss_riccitensor()
tracless_ricci_tensor
```

[27]:

$$\begin{bmatrix} \frac{1}{2a^2} & 0 & 0 & 0 \\ 0 & -\frac{1}{2a^3} & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & \frac{\sin^2(\theta)}{2a} \end{bmatrix}$$

[28]:
```
# Default type of the Traceless Ricci Tensor
trt.get_trclss_riccitensor_type()
```

[28]:
```
'dd'
```

[29]:
```
# Varying type 'dd' Traceless Ricci Tensor to 'uu'
trt.vary_trclss_riccitensor_type(tracless_ricci_tensor, 'uu')
```

[29]:
$$\begin{bmatrix} \frac{1}{2a^2} & 0 & 0 & 0 \\ 0 & -\frac{1}{2a} & 0 & 0 \\ 0 & 0 & \frac{1}{2a^4} & 0 \\ 0 & 0 & 0 & \frac{1}{2a^3\sin^2(\theta)} \end{bmatrix}$$

[30]:
```
trt.get_trclss_riccitensor_type()
```

[30]:
```
'uu'
```

7 Weyl Tensor

[31]:
```
# Obtaining the Weyl Tensor
wyl = WeylTensor(diag_comp, coord_sys)
weyl_tensor = wyl.get_weyltensor()
weyl_tensor
```

[31]:
$$\begin{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & -\frac{1}{3a^3} & 0 & 0 \\ \frac{1}{3a^3} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & \frac{1}{6} & 0 \\ 0 & 0 & 0 & 0 \\ -\frac{1}{6} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 & \frac{\sin^2(\theta)}{6a} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -\frac{\sin^2(\theta)}{6a} & 0 & 0 & 0 \end{bmatrix} \\ \begin{bmatrix} 0 & \frac{1}{3a^3} & 0 & 0 \\ -\frac{1}{3a^3} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{6a} & 0 \\ 0 & \frac{1}{6a} & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{\sin^2(\theta)}{6a^2} \\ 0 & 0 & 0 & 0 \\ 0 & \frac{\sin^2(\theta)}{6a^2} & 0 & 0 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 & -\frac{1}{6} & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{6} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{6a} & 0 \\ 0 & -\frac{1}{6a} & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -a\sin^2(\theta) \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{a\sin^2(\theta)}{3} \\ 0 & 0 & \frac{2a\sin^2(\theta)}{3} & 0 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 & 0 & -\frac{\sin^2(\theta)}{6a} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{\sin^2(\theta)}{6a} & 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\sin^2(\theta)}{6a^2} \\ 0 & 0 & 0 & 0 \\ 0 & -\frac{\sin^2(\theta)}{6a^2} & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{2a\sin^2(\theta)}{3} \\ 0 & 0 & \frac{a\sin^2(\theta)}{3} & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -a\sin^2(\theta) & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{bmatrix}$$

[32]:
```
# Default type of the Weyl Tensor
wyl.get_weyltensor_type()
```

[32]:
```
'dddd'
```

[33]:
```
# Varying type 'dddd' Weyl Tensor to 'uddd'
wyl.vary_weyltensor_type(weyl_tensor, 'uddd')
```

[33]:
$$\begin{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & \frac{1}{3a^3} & 0 & 0 \\ -\frac{1}{3a^3} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & -\frac{1}{6} & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{6} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 & -\frac{\sin^2(\theta)}{6a} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{\sin^2(\theta)}{6a} & 0 & 0 & 0 \end{bmatrix} \\ \begin{bmatrix} 0 & \frac{1}{3a^2} & 0 & 0 \\ -\frac{1}{3a^2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{6} & 0 \\ 0 & \frac{1}{6} & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{\sin^2(\theta)}{6a} \\ 0 & 0 & 0 & 0 \\ 0 & \frac{\sin^2(\theta)}{6a} & 0 & 0 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 & -\frac{1}{6a^2} & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{6a^2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{6a^3} & 0 \\ 0 & -\frac{1}{6a^3} & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{\sin^2(\theta)}{a} \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\sin^2(\theta)}{3a} \\ 0 & 0 & \frac{2\sin^2(\theta)}{3a} & 0 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 & 0 & -\frac{1}{6a^2} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{6a^2} & 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{6a^3} \\ 0 & 0 & 0 & 0 \\ 0 & -\frac{1}{6a^3} & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{2}{3} \\ 0 & 0 & \frac{1}{3} & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{bmatrix}$$

[34]:
```
wyl.get_weyltensor_type()
```

[34]:
```
'uddd'
```

8 Einstein Tensor

[35]:
```
# Obtaining the Einstein Tensor
et = EinsteinTensor(diag_comp, coord_sys)
einstein_tensor = et.get_einsteintensor()
einstein_tensor
```

[35]:
$$\begin{bmatrix} \frac{1}{a^2} & 0 & 0 & 0 \\ 0 & -\frac{1}{a^3} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

[36]:
```
# Default type of the Einstein Tensor
et.get_einsteintensor_type()
```

[36]:
```
'dd'
```

[37]:
```
# Varying type 'dd' Einstein Tensor to 'uu'
et.vary_einsteintensor_type(einstein_tensor, 'uu')
```

[37]:
$$\begin{bmatrix} \frac{1}{a^2} & 0 & 0 & 0 \\ 0 & -\frac{1}{a} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

[38]:
```
et.get_einsteintensor_type()
```

[38]:
```
'uu'
```