1 Metric $g_{\mu\nu}$ for spherically symmetric spacetime

[8]:
```
t, r, th, ph = symbols("t, r, \\theta, \\phi")
x1 = r * cos(ph) * sin(th)
x2 = r * sin(ph) * sin(th)
x3 = r * cos(th)

one = Rational(1)
eta = sp.diag(one, -one, -one, -one)
var = (t, r, th, ph)
J = Matrix([t, x1, x2, x3]).jacobian(var)
g = np.array(simplify(J.T *eta* J))

a = sp.Function("\\alpha", real=True)(r)
b = sp.Function("\\beta", real=True)(r)
g[0, 0] *= exp(2 * a)
g[1, 1] *= exp(2 * b)
g_inv = get_g_inv(g)

print_matrix(g)
print_matrix(g_inv)
```

$$\begin{bmatrix} e^{2\alpha(r)} & 0 & 0 & 0 \\ 0 & -e^{2\beta(r)} & 0 & 0 \\ 0 & 0 & -r^2 & 0 \\ 0 & 0 & 0 & -r^2\sin^2(\theta) \end{bmatrix}$$

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

[9]:
```
C = Christoffel(g, g_inv, var)
c = print_christoffel(C, var)
```

$$\Gamma^t_{\mu\nu} = \begin{bmatrix} 0 & \frac{d}{dr}\alpha(r) & 0 & 0 \\ \frac{d}{dr}\alpha(r) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\Gamma^{r}_{\mu\nu} = \begin{bmatrix} e^{2\alpha(r)}e^{-2\beta(r)}\frac{d}{dr}\alpha(r) & 0 & 0 & 0 \\ 0 & \frac{d}{dr}\beta(r) & 0 & 0 \\ 0 & 0 & -re^{-2\beta(r)} & 0 \\ 0 & 0 & 0 & -re^{-2\beta(r)}\sin^{2}(\theta) \end{bmatrix}$$

$$\Gamma^{\theta}_{\mu\nu} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{r} & 0 \\ 0 & \frac{1}{r} & 0 & 0 \\ 0 & 0 & 0 & -\sin(\theta)\cos(\theta) \end{bmatrix}$$

$$\Gamma^{\phi}_{\mu\nu} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{r} \\ 0 & 0 & 0 & \frac{\cos(\theta)}{\sin(\theta)} \\ 0 & \frac{1}{r} & \frac{\cos(\theta)}{\sin(\theta)} & 0 \end{bmatrix}$$

[10]:
```
Rie = Riemann_tensor(C, var)
Ricci = contract(Rie, num_indx=4, upper=1, indx=(0, 2))

for i in range(4):
    print_scalar(Ricci[i, i].factor())
```

$$\frac{\left(r\left(\frac{d}{dr}\alpha(r)\right)^{2} - r\frac{d}{dr}\alpha(r)\frac{d}{dr}\beta(r) + r\frac{d^{2}}{dr^{2}}\alpha(r) + 2\frac{d}{dr}\alpha(r)\right)e^{2\alpha(r)}e^{-2\beta(r)}}{r}$$

$$-\frac{r\left(\frac{d}{dr}\alpha(r)\right)^{2} - r\frac{d}{dr}\alpha(r)\frac{d}{dr}\beta(r) + r\frac{d^{2}}{dr^{2}}\alpha(r) - 2\frac{d}{dr}\beta(r)}{r}$$

$$-\left(r\frac{d}{dr}\alpha(r) - r\frac{d}{dr}\beta(r) - e^{2\beta(r)} + 1\right)e^{-2\beta(r)}$$

$$-\left(r\frac{d}{dr}\alpha(r) - r\frac{d}{dr}\beta(r) - e^{2\beta(r)} + 1\right)e^{-2\beta(r)}\sin^{2}(\theta)$$

[11]:
```
R = contract(Ricci, g_inv=g_inv, upper=0).simplify()
print_scalar(R)
```

$$\frac{2\left(r^2\left(\frac{d}{dr}\alpha(r)\right)^2 - r^2\frac{d}{dr}\alpha(r)\frac{d}{dr}\beta(r) + r^2\frac{d^2}{dr^2}\alpha(r) + 2r\frac{d}{dr}\alpha(r) - 2r\frac{d}{dr}\beta(r) - e^{2\beta(r)} + 1\right)e^{-2\beta(r)}}{r^2}$$

[12]:
```
G = Ricci - Rational(1, 2) * R * g
for i in range(4):
    G[i, i] = G[i, i].simplify().factor()
    print_scalar(G[i, i])
```

$$\frac{\left(2r\frac{d}{dr}\beta(r) + e^{2\beta(r)} - 1\right)e^{2\alpha(r)}e^{-2\beta(r)}}{r^2}$$

$$\frac{2r\frac{d}{dr}\alpha(r) - e^{2\beta(r)} + 1}{r^2}$$

$$r\left(r\left(\frac{d}{dr}\alpha(r)\right)^2 - r\frac{d}{dr}\alpha(r)\frac{d}{dr}\beta(r) + r\frac{d^2}{dr^2}\alpha(r) + \frac{d}{dr}\alpha(r) - \frac{d}{dr}\beta(r)\right)e^{-2\beta(r)}$$

$$r\left(r\left(\frac{d}{dr}\alpha(r)\right)^2 - r\frac{d}{dr}\alpha(r)\frac{d}{dr}\beta(r) + r\frac{d^2}{dr^2}\alpha(r) + \frac{d}{dr}\alpha(r) - \frac{d}{dr}\beta(r)\right)e^{-2\beta(r)}\sin^2(\theta)$$

1.0.1 Stress-energy tensor $T_{\mu\nu}$ for perfect fluid

[13]:
```
p = sp.Function("p")(r)
u = sp.Function("u")(r)

UU = np.zeros((4, 4), dtype=sp.Rational)
UU[0, 0] = exp(2 * a)

T = (p + u) * UU - p * g
for i in range(4):
    T[i, i] = T[i, i].simplify()
print_matrix(T)
```

$$\begin{bmatrix} u(r)e^{2\alpha(r)} & 0 & 0 & 0 \\ 0 & p(r)e^{2\beta(r)} & 0 & 0 \\ 0 & 0 & r^2p(r) & 0 \\ 0 & 0 & 0 & r^2p(r)\sin^2(\theta) \end{bmatrix}$$

2 Einstin's field equations

$R_{\mu\nu} - \frac{1}{2}Rg_{\mu\nu} = 8\pi GT_{\mu\nu}$

[14]:
```
G_newton = sp.Symbol("G")

eq = []
for i in range(len(G)):
    eq.append((G[i, i] - 8 * pi * G_newton * T[i, i]).simplify())

# Some manual simplification
Rtt = sp.Symbol("R_{\\theta \\theta}")
eq[0] = eq[0] * r**2 / exp(2 * a)/exp(-2*b ) * (-1 )
eq[1] = eq[1] * r**2 * (-1)
eq[2] = eq[2] / r / exp(-2*b)
eq[3] = eq[3].subs(eq[2], Rtt)
for i in range(len(G)):
    print_eq(eq[i].simplify())
```

$$8\pi Gr^2u(r)e^{2\beta(r)} - 2r\frac{d}{dr}\beta(r) - e^{2\beta(r)} + 1 = 0$$

$$8\pi Gr^2p(r)e^{2\beta(r)} - 2r\frac{d}{dr}\alpha(r) + e^{2\beta(r)} - 1 = 0$$

$$-8\pi Grp(r)e^{2\beta(r)} + r\left(\frac{d}{dr}\alpha(r)\right)^2 - r\frac{d}{dr}\alpha(r)\frac{d}{dr}\beta(r) + r\frac{d^2}{dr^2}\alpha(r) + \frac{d}{dr}\alpha(r) - \frac{d}{dr}\beta(r) = 0$$

$$R_{\theta\theta}re^{-2\beta(r)}\sin^2(\theta) = 0$$

Define $e^{2\beta} = [1 - 2Gm(r)/r]^{-1}$

[15]:
```
m = sp.Function("m", Real=True)(r)
f = (1 - 2 * G_newton * m / r)**(-1)
eq1 = (eq[0] * exp(- 2 *a)).simplify().subs(b, Rational(1, 2) * log(f)).
 ↪simplify().expand().simplify()
s = sp.solve(eq1, m.diff(r))
eq1 = m.diff(r) - s[0]
```

Use $\nabla_\mu T^{\mu r} = 0 \implies (p+\rho)\partial_r \alpha = -\partial_r p$.

[16]:
```
eq2 = (eq[1] * r**2).subs(exp(2 * b), f).simplify()
s = sp.solve(eq2, a.diff(r))
eq2 = a.diff(r) - s[0]
eq2 = ((a.diff(r) - s[0]).subs(a.diff(r), - p.diff(r) / (p + u))*(p + u)).
 ↪simplify()
s = sp.solve(eq2, p.diff())
eq2 = p.diff(r) - s[0].factor()
```

The TOV-equation and equation for $m(r)$, both expressions are equal to 0.

[17]:
```
print_eq(eq1)
print_eq(eq2)
```

$$-4\pi r^2 u(r) + \frac{d}{dr} m(r) = 0$$

$$\frac{G\left(4\pi r^3 p(r) + m(r)\right)\left(p(r) + u(r)\right)}{r\left(-2Gm(r) + r\right)} + \frac{d}{dr} p(r) = 0$$