a) The metric

We have a static, isotropic metric. That is, there is no t-dependence and it is invariant under rotation. Denote our coordinates with $x^\mu = (t, r, \theta, \phi)$. We assume 1+3 dimensions, one temporal and three spatial. Isotropic in spatial part, means we have a SO(3) group (ie. a "sphere"). Thus, the metric must contain

$$g_\Omega = d\theta^2 + \sin^2\theta d\phi^2.$$

We write our metric as

$$ds^2 = g(r) + \gamma(r) g_\Omega,$$

where there is no t-dependence due to it being static. Also, the angular dependence is entirely contained in g_Ω, to not break isotropy. Thus, g is on the form

$$g(r) = g_{\mu\nu} dx^\mu dx^\nu, \quad \mu, \nu \in \{t, r\}.$$

Any cross terms between dt and dr will break time reversal, as one will get something on the form

$$ds^2 = dt^2 + dtdr + dr^2 = (\frac{dt}{dr}^2 + \frac{dt}{dr} + 1) dr^2.$$

As the second term is not quadratic in dt, it changes sign on time reversal, breaking the symmetry. Thus, there can be no cross terms. Our metric is thus

$$ds^2 = A(r) dt^2 - B(r) dr^2 + \gamma(r)(d\theta^2 + \sin^2\theta d\phi^2).$$

Signs were chosen by convention. As we have no constraints on "r", we may choose it as we wish. We choose it such that

$$ds^2 = A(r) dt^2 - B(r) dr^2 - r^2(d\theta^2 + \sin^2\theta d\phi^2).$$

b) Ricci tensor

We will here use the framework "differentialGeometry3" provided through the course webpage, `http://web.phys.ntnu.no/~mika/QF/software.html`.

```python
import numpy as np
import sympy
from differentialGeometry3 import computeGeometry, printGeometry,
    determinantg, ginv
from IPython.display import display, Math

def displayNonZero(T):
    """Displays the non-zero components of the tensor T
    Also assumes to be inside iPython or Jupyter, to
    use the `display` function"""
    for index, element in np.ndenumerate(T):
        if element != 0:
```

```
            print(index)
            display(sympy.simplify(element))


# Define our coordinates
t, r, theta, phi = sympy.symbols('t r theta phi')
coords = np.array([t, r, theta, phi])
A = sympy.Function('A')(r)
B = sympy.Function('B')(r)

# Construct the tensor
g_ = np.zeros((4, 4),dtype=object)
g_[0,0] = A
g_[1,1] = -B
g_[2,2] = -r**2
g_[3,3] = -r**2 * sympy.sin(theta)**2

geometry = computeGeometry(g_,coords)

# Ricci
r4i_ = geometry[7]

# Ricci is diagonal
for ele in np.diag(r4i_):
    display(sympy.simplify(ele))
    print("———")
```

Which outputs

$$\frac{\frac{d^2}{dr^2}A(r)}{2A(r)} - \frac{\frac{d}{dr}A(r)\frac{d}{dr}B(r)}{4A(r)B(r)} - \frac{\left(\frac{d}{dr}A(r)\right)^2}{4A^2(r)} - \frac{\frac{d}{dr}B(r)}{rB(r)}$$

—

$$\frac{\frac{d^2}{dr^2}A(r)}{2A(r)} - \frac{\frac{d}{dr}A(r)\frac{d}{dr}B(r)}{4A(r)B(r)} - \frac{\left(\frac{d}{dr}A(r)\right)^2}{4A^2(r)} - \frac{\frac{d}{dr}B(r)}{rB(r)}$$

—

$$-\frac{r\frac{d}{dr}B(r)}{2B^2(r)} + \frac{r\frac{d}{dr}A(r)}{2A(r)B(r)} - 1 + \frac{1}{B(r)}$$

—

$$\frac{\left(-rA(r)\frac{d}{dr}B(r) + rB(r)\frac{d}{dr}A(r) - 2\left(B(r)-1\right)A(r)B(r)\right)\sin^2(\theta)}{2A(r)B^2(r)}$$

Which as one can see, with a trivial amount of algebra, is exactly what we are supposed to show.

c) Maxwell equation

From the program mentioned above, we have $|g| = -r^4 AB\sin^2\theta$. The inhomogeneous Maxwell equation is

$$\nabla_\mu F^{\mu\nu} = \frac{1}{\sqrt{|g|}}\partial\mu\left(\sqrt{|g|}F^{\mu\nu}\right) = j^\nu.$$

We have that $F_{\mu\nu} = -E_\mu$, and from the isometric property $E_\theta = E_\phi = 0$ (which actually follows from the hairy ball theorem). $j^0 = \rho$, where ρ is the charge density, which is zero everywhere except at the point charge. Thus

$$\partial_\mu \left(\sqrt{|g|} F^{\mu 0} \right) = 0.$$

Given that E_r is the only non-zero component of E, and the relation between F and E, we get

$$\partial_r \left(i r^2 \sin\theta \sqrt{AB} F^{r0} \right) = 0.$$

Which we may write more conveniently as

$$\partial_r \left(r^2 \sqrt{AB} F^{r0} \right) = 0,$$

so that

$$r^2 \sqrt{AB} F^{r0} = C$$

where C is some constant. Using

$$F^{r0} = g^{rr} F_{r0} g^{00} = -\frac{F_{r0}}{AB} = \frac{E_r}{AB},$$

where we used that g is diagonal, we conclude that

$$r^2 \sqrt{AB} F^{r0} = C \Rightarrow E_r = \frac{C\sqrt{AB}}{r^2}.$$

Lastly remains determining C. As we move to $r \to \infty$, we must approach Minkowski space, that is $A = B = 1$. Thus, applying Gauss's law at large r, we simply get

$$\int dS E(r) = Q,$$

and we identify that $E = \frac{Q}{4\pi r^2}$. Comparing with our expression for E, we see that $C = \frac{Q}{4\pi}$.

d) Electromagnetic stress tensor and Einstein equations

Again, we make use of the aforementioned framework. The stress tensor is defined as

$$T^{\mu\nu} = -F^{\mu\alpha} F^{\nu}_{\alpha} + \frac{1}{4} g^{\mu\nu} F_{\alpha\beta} F^{\alpha\beta}.$$

We implement it in Python

```python
def stressTensor(F, g_):
  """Compute stress tensor T from EM-field F"""
  g = ginv(g_)
  d = g_[:,0].size
  T = np.zeros((d,d),dtype=object)
  for mu in range(d):
      for nu in range(d):
          T[mu, nu] = F[mu, :] @ (g_ @ F)[:, nu] - sympy.Rational(1, 4) * g[mu, nu] * np.tensordot(g_ @ F @ g_, F)

          return T

def lower(T):
  """Lowers the indices of the (upper) tensor T"""
  return g_ @ T @ g_
```

Here, we also defined the function lower because we need the stress tensor with lower indices. Using our knowledge about E, and that there is no B-field, we construct F.

$$F_{\mu\nu} = \begin{bmatrix} 0 & E_r & 0 & 0 \\ -E_r & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Or, in Python code,

```python
F = np.zeros(g_.shape, dtype=object)
Q = sympy.symbols('Q')
E = sympy.sqrt(A * B) * Q / (4*sympy.pi*r**2)

F[1, 0] = - E / (A * B)
F[0, 1] = -F[1, 0]

T = stressTensor(F, g_)
T_ = lower(T)
displayNonZero(T_)
```

Which outputs

(0,0)

$$-\frac{3Q^2 A(r)}{32\pi^2 r^4}$$

(1,1)

$$\frac{3Q^2 B(r)}{32\pi^2 r^4}$$

(2,2)

$$\frac{Q^2}{32\pi^2 r^2}$$

(3,3)

$$\frac{Q^2 \sin^2(\theta)}{32\pi^2 r^2}$$

Now, to show that the Einstein equation

$$G_{\mu\nu} = -\kappa T_{\mu\nu}$$

reduces to

$$R_{\mu\nu} = -\kappa T_{\mu\nu}$$

we must employ a trick. Firstly, note

$$G_{\mu\nu} = R_{\mu\nu} - \frac{1}{2} R g_{\mu\nu}.$$

Consider

$$G_\mu^\nu = R_\mu^\nu - \frac{1}{2} R g_\mu^\nu = -\kappa T_\mu^\nu.$$

If we take the trace on both sides, and denote by only the letter of the tensor its trace, we get

$$R - 2R = -\kappa T = 0.$$

We here used that the trace of g is 4 ($g_\mu^\nu = \delta_\mu^\nu$), and that the electromagnetic stress tensor is traceless. Thus, $R = 0$, and the Einstein equation reduces as we wanted to show. Using our framework, we may find the explicit form of the (non-trivial) Einstein equations.

```
k = sympy.Symbol("k")

for i in range(4):
    print(f``({i}, {i})'')
    display(sympy.Eq(ricci_[i, i], -k * T_[i, i]))
```

Giving

(0,0)

$$-\frac{\frac{d^2}{dr^2}A(r)}{2B(r)} + \frac{\frac{d}{dr}A(r)\frac{d}{dr}B(r)}{4B^2(r)} + \frac{\left(\frac{d}{dr}A(r)\right)^2}{4A(r)B(r)} - \frac{\frac{d}{dr}A(r)}{rB(r)} = \frac{3Q^2kA(r)}{32\pi^2 r^4}$$

(1,1)

$$\frac{\frac{d^2}{dr^2}A(r)}{2A(r)} - \frac{\frac{d}{dr}A(r)\frac{d}{dr}B(r)}{4A(r)B(r)} - \frac{\left(\frac{d}{dr}A(r)\right)^2}{4A^2(r)} - \frac{\frac{d}{dr}B(r)}{rB(r)} = -\frac{3Q^2kB(r)}{32\pi^2 r^4}$$

(2,2)

$$-\frac{r\frac{d}{dr}B(r)}{2B^2(r)} + \frac{r\frac{d}{dr}A(r)}{2A(r)B(r)} - 1 + \frac{1}{B(r)} = -\frac{Q^2k}{32\pi^2 r^2}$$

(3,3)

$$-\frac{r\sin^2(\theta)\frac{d}{dr}B(r)}{2B^2(r)} + \frac{r\sin^2(\theta)\frac{d}{dr}A(r)}{2A(r)B(r)} - \frac{(B(r)-1)\sin^2(\theta)}{B(r)} = -\frac{Q^2k\sin^2(\theta)}{32\pi^2 r^2}$$

e) Finding the relation AB

We notice that taking

$$BR_{00} + AR_{11} = -\kappa(BT_{00} + AT_{11})$$

vastly simplifies our expression, as the right hand site is zero. Thus, after computing the LHS

$$-\frac{A(r)\frac{d}{dr}B(r)}{rB(r)} - \frac{\frac{d}{dr}A(r)}{r} = 0.$$

Multiplying away r and raising the B, we recognize this as simply

$$\frac{d}{dr}(AB) = 0.$$

As we may rescale our t as we want, we get

$$AB = const = 1.$$

f) Determining A and B

We have

$$R_{22} = \frac{1}{B} - 1 + \frac{r}{2B}(\frac{A'}{A} - \frac{B'}{B}) = -\kappa T_{22}.$$

Inserting $A = \frac{1}{B}$ and solving using our framework, we get

$$B(r) = \frac{32\pi^2 r^2}{C_1 r + Q^2 k + 32\pi^2 r^2}$$
$$A(r) = \frac{C_1}{32\pi^2 r} + \frac{Q^2 k}{32\pi^2 r^2} + 1$$

We expect to regain the Schwarzschild solution as $Q \to 0$, so we have

$$\lim_{Q\to 0} A(r) = \frac{C_1}{32\pi^2 r} + 1 = 1 - \frac{2GM}{r}.$$

We here inserted g_{00} for the Schwarzschild metric. Thus, $C_1 = -2GM \cdot 32\pi^2 = -64GM\pi^2$. And we get, after inserting $\kappa = 8\pi G$,

$$A(r) = 1 - \frac{2GM}{r} + \frac{GQ^2}{4\pi r^2}$$
$$B(r) = \left(1 - \frac{2GM}{r} + \frac{GQ^2}{4\pi r^2}\right)^{-1}$$

g) Singularities

Similarly to Schwarzschild, we get a singularity at $r = 0$ from g_{tt} and one in g_{rr} from the denominator going to zero. Let us consider the second one, which comes from B. Since $B = A^{-1}$, we need to find the zeros of A. Simply solving using our framework, and with the help of sympy, we get that the zero points of A are

$$r = GM\left(1 \pm \sqrt{1 - \frac{Q^2}{4\pi GM^2}}\right).$$

We are happy to to see that for $Q \to 0$, this reduces to $r = 0$ and $r = 2GM$, which is as expected.