The Mathematica package RGTC

There are several useful packages[1] to compute curvature invariants, covariant derivatives, and so on, for arbitrary metrics. The one I normally use is called *Riemann Geometry & Tensor Calculus (RGTC)* and can be downloaded for free from

`http://www.inp.demokritos.gr/~sbonano/RGTC/`

You need to copy the `EDCRGTCcode.m` file in your main `Mathematica` directory and then recall it at the beginning of your notebook using

```
<<EDCRGTCcode.m
```

After you do this operation you need to input the coordinates and metric. For example, the AdS_4 metric with radius L

$$ds^2 = -\left(\frac{r^2}{L^2}+1\right)dt^2 + \frac{dr^2}{\frac{r^2}{L^2}+1} + r^2 d\Omega_2^2$$

would be given by

```
Coordinates = {t, r, θ, φ}
Metric = DiagonalMatrix[{-(r^2/L^2+1), 1/(r^2/L^2+1), r^2, r^2 Sin[θ]^2}]
```

Now run the command

```
RGtensors[Metric,Coordinates]
```

This will automatically compute for you a bunch of curvature invariants. There are commands that help contracting indices. For example:

- `gdd` is the metric, $g_{\mu\nu}$
- `gUU` is the inverse metric, $g^{\mu\nu}$
- `Rdddd` is the Riemann tensor with lower indices, $R_{\mu\nu\rho\sigma}$
- `Rdd` is the Ricci tensor with lower indices, $R_{\mu\nu}$
- `R` is the scalar curvature, R (note that `R` is protected, so don't call anything else with this symbol in your notebook)

[1] Other excellent packages are `diffeo.m` by Matthew Headrick
`http://people.brandeis.edu/~headrick/Mathematica/`
and `Ricci.m` by John Lee
`http://www.math.washington.edu/~lee/Ricci/`.

- `covD[ ]` is the covariant derivative with lower index, ∇_μ
- `multiDot[A,B,{i,j},{m,n},...]` sums the i-th index of the A tensor with the j-th index of the B tensor, the m-th index of A with the n-th index of B, etc.

and so on.

For example, if you want to implement

$$g^{\mu\nu}\nabla_\mu\nabla_\nu f(t,\mathbf{r})\,,$$

where f is a scalar function you'll need to use

```
multiDot[gUU,covD[covD[f[t, r, θ, ϕ]]],{1,1},{2,2}]
```

while for

$$g^{\mu\nu}\nabla_\mu f(t,\mathbf{r})\nabla_\nu f(t,\mathbf{r})$$

you may simply use

```
covD[f[t, r, θ, ϕ]].gUU.covD[f[t, r, θ, ϕ]]
```

since the lower dot `.` means matrix multiplication. The outer product of forms, e.g.

$$F_\mu F_\nu$$

is given by

```
Outer[Times,Fd,Fd]
```

In the package documentation there are some neat examples you can have a look at. In your first problem set you'll be asked to use this package to verify that specific metrics are solutions to Einstein equations of motion.