Load Required Packages

In[1]:=
```
<< xAct`xPert`
<< xAct`xCoba`
```

```
------------------------------------------------------------
Package xAct`xPerm`  version 1.2.3, {2015, 8, 23}
CopyRight (C) 2003-2020, Jose M. Martin-Garcia, under the General Public License.
Connecting to external MinGW executable...
Connection established.
------------------------------------------------------------
Package xAct`xTensor`  version 1.1.5, {2021, 2, 28}
CopyRight (C) 2002-2021, Jose M. Martin-Garcia, under the General Public License.
------------------------------------------------------------
Package xAct`xPert`  version 1.0.6, {2018, 2, 28}
CopyRight (C) 2005-2020, David Brizuela, Jose M. Martin-Garcia
  and Guillermo A. Mena Marugan, under the General Public License.
------------------------------------------------------------
These packages come with ABSOLUTELY NO WARRANTY; for details type
  Disclaimer[]. This is free software, and you are welcome to redistribute
  it under certain conditions. See the General Public License for details.
------------------------------------------------------------
** Variable $PrePrint assigned value ScreenDollarIndices
** Variable $CovDFormat changed from Prefix to Postfix
** Option AllowUpperDerivatives of ContractMetric changed from False to True
** Option MetricOn of MakeRule changed from None to All
** Option ContractMetrics of MakeRule changed from False to True
------------------------------------------------------------
Package xAct`xCoba`  version 0.8.6, {2021, 2, 28}
CopyRight (C) 2005-2021, David Yllanes
  and Jose M. Martin-Garcia, under the General Public License.
------------------------------------------------------------
These packages come with ABSOLUTELY NO WARRANTY; for details type
  Disclaimer[]. This is free software, and you are welcome to redistribute
  it under certain conditions. See the General Public License for details.
------------------------------------------------------------
```

In this notebook we try to compute the expressions of components of Christoffel symbols, Riemann curvature tensor etc for a perturbed matric.
the metric is composed of a background metric and its perturbations

In[3]:=
```
$PrePrint = ScreenDollarIndices
```

Out[3]=
```
ScreenDollarIndices
```

Abstract part

First we need a 4D manifold:

```
In[4]:= DefManifold[M, 4, {α, β, γ, μ, ν, λ, ρ, σ, η}]
```

```
** DefManifold: Defining manifold M.

** DefVBundle: Defining vbundle TangentM.
```

Defining an abstract metric

```
In[5]:= DefMetric[-1, background[-μ, -ν], cd, PrintAs → "g"]
```

```
** DefTensor: Defining symmetric metric tensor background[-μ, -ν].

** DefTensor: Defining antisymmetric tensor epsilonbackground[-α, -β, -γ, -η].

** DefTensor: Defining tetrametric Tetrabackground[-α, -β, -γ, -η].

** DefTensor: Defining tetrametric Tetrabackground† [-α, -β, -γ, -η].

** DefCovD: Defining covariant derivative cd[-μ].

** DefTensor: Defining vanishing torsion tensor Torsioncd[α, -β, -γ].

** DefTensor: Defining symmetric Christoffel tensor Christoffelcd[α, -β, -γ].

** DefTensor: Defining Riemann tensor Riemanncd[-α, -β, -γ, -η].

** DefTensor: Defining symmetric Ricci tensor Riccicd[-α, -β].

** DefCovD:  Contractions of Riemann automatically replaced by Ricci.

** DefTensor: Defining Ricci scalar RicciScalarcd[].

** DefCovD:  Contractions of Ricci automatically replaced by RicciScalar.

** DefTensor: Defining symmetric Einstein tensor Einsteincd[-α, -β].

** DefTensor: Defining Weyl tensor Weylcd[-α, -β, -γ, -η].

** DefTensor: Defining symmetric TFRicci tensor TFRiccicd[-α, -β].

** DefTensor: Defining Kretschmann scalar Kretschmanncd[].

** DefCovD:  Computing RiemannToWeylRules for dim 4

** DefCovD:  Computing RicciToTFRicci for dim 4

** DefCovD:  Computing RicciToEinsteinRules for dim 4

** DefTensor: Defining weight +2 density Detbackground[]. Determinant.
```

Construct a perturbative structure:

```
In[6]:= DefParameter[ε]
```

```
** DefParameter: Defining parameter ε.
```

```
In[7]:= DefMetricPerturbation[background, pert, ε, PrintAs → "h"]
```

```
** DefTensor: Defining tensor pert[LI[order], -α, -β].
```

```
In[8]:= Unprotect[IndexForm];
IndexForm[LI[x_]] := ColorString[ToString[x], Red]
```

We will be interested in Christoffel symbols,Riemann Curvature tensor, Ricci tensor, Ricci scalar for this metric. here 1 stands for order 1. similar methods can be used for obtaining the expressions for higher orders in the same manner

Christoffel symbol (order 1):

In[10]:=
```
C1[α, -β, -γ] =
 ToCanonical[Perturbed[Christoffelcd[α, -β, -γ], 1] // ExpandPerturbation //
   SeparateMetric[background], UseMetricOnVBundle → None]
```

Out[10]= $\Gamma[\nabla]^{\alpha}{}_{\beta\gamma} + \frac{1}{2}\varepsilon\, g^{\alpha\eta}\, h^{1}{}_{\gamma\eta;\beta} + \frac{1}{2}\varepsilon\, g^{\alpha\eta}\, h^{1}{}_{\beta\eta;\gamma} - \frac{1}{2}\varepsilon\, g^{\alpha\eta}\, h^{1}{}_{\beta\gamma;\eta}$

Similarly we can define the abstract expressions for the other tensors we are interested in:

Riemann Curvature tensor(order 1):

In[11]:=
```
Riem[-μ, -ν, -ρ, -σ] =
 ToCanonical[Perturbed[Riemanncd[-μ, -ν, -ρ, -σ], 1] // ExpandPerturbation //
   SeparateMetric[background], UseMetricOnVBundle → None]
```

Out[11]= $\varepsilon\, g^{\alpha\beta}\, h^{1}{}_{\sigma\alpha}\, R[\nabla]_{\mu\nu\rho\beta} + R[\nabla]_{\mu\nu\rho\sigma} - \frac{1}{2}\varepsilon\, h^{1}{}_{\rho\sigma;\nu;\mu} - \frac{1}{2}\varepsilon\, h^{1}{}_{\nu\sigma;\rho;\mu} + \frac{1}{2}\varepsilon\, h^{1}{}_{\nu\rho;\sigma;\mu} + \frac{1}{2}\varepsilon\, h^{1}{}_{\rho\sigma;\mu;\nu} + \frac{1}{2}\varepsilon\, h^{1}{}_{\mu\sigma;\rho;\nu} - \frac{1}{2}\varepsilon\, h^{1}{}_{\mu\rho;\sigma;\nu}$

Ricci Tensor (order 1) :

In[12]:=
```
Ricci1[-μ, -ν] = ToCanonical[Perturbed[Riccicd[-μ, -ν], 1] // ExpandPerturbation //
   SeparateMetric[background], UseMetricOnVBundle → None]
```

Out[12]= $R[\nabla]_{\mu\nu} - \frac{1}{2}\varepsilon\, g^{\alpha\beta}\, h^{1}{}_{\mu\nu;\alpha;\beta} + \frac{1}{2}\varepsilon\, g^{\alpha\beta}\, h^{1}{}_{\nu\alpha;\mu;\beta} + \frac{1}{2}\varepsilon\, g^{\alpha\beta}\, h^{1}{}_{\mu\alpha;\nu;\beta} - \frac{1}{2}\varepsilon\, g^{\alpha\beta}\, h^{1}{}_{\alpha\beta;\nu;\mu}$

Ricci Scalar (order 1):

In[13]:=
```
RS = ToCanonical[Perturbed[RicciScalarcd[], 1] // ExpandPerturbation //
   SeparateMetric[background], UseMetricOnVBundle → None]
```

Out[13]= $-\varepsilon\, g^{\alpha\beta}\, g^{\gamma\eta}\, h^{1}{}_{\alpha\gamma}\, R[\nabla]_{\beta\eta} + R[\nabla] + \varepsilon\, g^{\alpha\beta}\, g^{\gamma\eta}\, h^{1}{}_{\alpha\gamma;\beta;\eta} - \varepsilon\, g^{\alpha\beta}\, g^{\gamma\eta}\, h^{1}{}_{\alpha\beta;\gamma;\eta}$

Component Part

Since we will be working with coordinates, we need to define our chart, it will be named as ch, the coordinates are numerated as {0,1,2,3},they will be t,x,y,and z (notice that we need to add [] after each coordinate as they will be defined as scaler fields).

In[14]:=
```
DefChart[ch, M, {0, 1, 2, 3}, {t[], x[], y[], z[]}, ChartColor → Purple]
```

```
** DefChart: Defining chart ch.
** DefTensor: Defining coordinate scalar t[].
** DefTensor: Defining coordinate scalar x[].
** DefTensor: Defining coordinate scalar y[].
** DefTensor: Defining coordinate scalar z[].
** DefMapping: Defining mapping ch.
** DefMapping: Defining inverse mapping ich.
** DefTensor: Defining mapping differential tensor dich[-a, ichα].
** DefTensor: Defining mapping differential tensor dch[-α, cha].
** DefBasis: Defining basis ch. Coordinated basis.
** DefCovD: Defining parallel derivative PDch[-α].
** DefTensor: Defining vanishing torsion tensor TorsionPDch[α, -β, -γ].
** DefTensor: Defining symmetric Christoffel tensor ChristoffelPDch[α, -β, -γ].
** DefTensor: Defining vanishing Riemann tensor RiemannPDch[-α, -β, -γ, η].
** DefTensor: Defining vanishing Ricci tensor RicciPDch[-α, -β].
** DefTensor: Defining antisymmetric +1 density etaUpch[α, β, γ, η].
** DefTensor: Defining antisymmetric -1 density etaDownch[-α, -β, -γ, -η].
```

For the 4D metric, we’ll need to define the required scalar functions, constants etc

```
In[15]:= DefScalarFunction[ϕ]
         ** DefScalarFunction: Defining scalar function ϕ.

In[16]:= $Assumptions ϕ[x, y, z] ∈ Reals
Out[16]= ℝ True ∈ ϕ[x, y, z]

In[17]:= DefConstantSymbol[speedc, PrintAs → "C"]
         ** DefConstantSymbol: Defining constant symbol speedc.

In[18]:= DefConstantSymbol[mass, PrintAs → "M"]
         ** DefConstantSymbol: Defining constant symbol mass.
```

Defining the elements of background metric (Here I have taken Minkowski metric as background)

```
In[19]:= backgroundmetric = CTensor[DiagonalMatrix[{-1, 1, 1, 1}], {-ch, -ch}]
Out[19]= CTensor[{{-1, 0, 0, 0}, {0, 1, 0, 0}, {0, 0, 1, 0}, {0, 0, 0, 1}}, {-ch, -ch}, 0]

In[20]:= backgroundmetric[-μ, -ν]
```

Out[20]= $\boxed{\begin{matrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}}_{\mu\nu}$

```
In[21]:= MetricCompute[backgroundmetric, ch, All]
```

In[22]:=
```
bgcd = CovDOfMetric[backgroundmetric]
```
Out[22]= PDch

Defining the perturbations

In[23]:=
```
$LargeComponentSize = 2000;
```
In[24]:=
```
perturbations = CTensor[
  DiagonalMatrix[{-2 * ϕ[x[], y[], z[]] / speedc^2, -2 * ϕ[x[], y[], z[]] / speedc^2,
    -2 * ϕ[x[], y[], z[]] / speedc^2, -2 * ϕ[x[], y[], z[]] / speedc^2}], {-ch, -ch}]
```
Out[24]= $\text{CTensor}\left[\left\{\left\{-\frac{2\,\phi[x,y,z]}{c^2},0,0,0\right\},\left\{0,-\frac{2\,\phi[x,y,z]}{c^2},0,0\right\},\left\{0,0,-\frac{2\,\phi[x,y,z]}{c^2},0\right\},\left\{0,0,0,-\frac{2\,\phi[x,y,z]}{c^2}\right\}\right\},\{-ch,-ch\},0\right]$

In[25]:=
```
perturbations[-μ, -ν]
```
Out[25]= $\begin{pmatrix} -\frac{2\,\phi[x,y,z]}{c^2} & 0 & 0 & 0 \\ 0 & -\frac{2\,\phi[x,y,z]}{c^2} & 0 & 0 \\ 0 & 0 & -\frac{2\,\phi[x,y,z]}{c^2} & 0 \\ 0 & 0 & 0 & -\frac{2\,\phi[x,y,z]}{c^2} \end{pmatrix}_{\mu\nu}$

So , now we have the expressions for background metric and perturbations. Any general background metric and perturbations can be used, here I have used a very simple case.

So, the next step is to use the defined background metric and perturbations in the expression of Christoffel Symbols, Riemann Curvature tensor etc obtained using xPert in the beginning of this notebook.
lets go back to those expressions: (Ricci scalar)

In[26]:=
```
RS
```
Out[26]= $-\varepsilon\, g^{\alpha\beta}\, g^{\gamma\eta}\, h^{1}{}_{\alpha\gamma}\, R[\nabla]_{\beta\eta}+R[\nabla]+\varepsilon\, g^{\alpha\beta}\, g^{\gamma\eta}\, h^{1}{}_{\alpha\gamma;\beta;\eta}-\varepsilon\, g^{\alpha\beta}\, g^{\gamma\eta}\, h^{1}{}_{\alpha\beta;\gamma;\eta}$

We need to replace abstract indices with component values. right now this metric g and perturbations h in the above expression and the CTensors background and perturbations we have defined using xCoba are different. We need to specify these as the same objects and compute the expressions.

In[27]:=
```
rules = {
   background[-μ_Symbol, -ν_Symbol] → backgroundmetric[-μ, -ν],
   background[μ_Symbol, ν_Symbol] → Inv[backgroundmetric][μ, ν],
   cd → bgcd,
   Riccicd[inds__] → Ricci[bgcd][inds],
   RicciScalarcd[] → RicciScalar[bgcd][],
   Einsteincd[inds__] → Einstein[bgcd][inds],
   Christoffelcd[inds__] → Christoffel[bgcd][inds],
   Riemanncd[inds__] → Riemann[bgcd][inds],
   pert[LI[1], inds__] → perturbations[inds]
  };
```

In[28]:=
```
$CVSimplify = Expand;
```

Christoffel Symbol:

In[29]:=
```
PChristoffel1 = Collect[C1[α, -β, -γ] /. rules, ε, Simplify]
```

Out[29]=

$$\Gamma[\mathcal{D}]^{\alpha}{}_{\beta\gamma} + \left(\begin{array}{cccc}
\begin{pmatrix} 0 \\ \frac{\varepsilon\,\phi^{(1,0,0)}[x,y,z]}{c^2} \\ \frac{\varepsilon\,\phi^{(0,1,0)}[x,y,z]}{c^2} \\ \frac{\varepsilon\,\phi^{(0,0,1)}[x,y,z]}{c^2} \end{pmatrix} &
\begin{pmatrix} \frac{\varepsilon\,\phi^{(1,0,0)}[x,y,z]}{c^2} \\ 0 \\ 0 \\ 0 \end{pmatrix} &
\begin{pmatrix} \frac{\varepsilon\,\phi^{(0,1,0)}[x,y,z]}{c^2} \\ 0 \\ 0 \\ 0 \end{pmatrix} &
\begin{pmatrix} \frac{\varepsilon\,\phi^{(0,0,1)}[x,y,z]}{c^2} \\ 0 \\ 0 \\ 0 \end{pmatrix} \\
\begin{pmatrix} \frac{\varepsilon\,\phi^{(1,0,0)}[x,y,z]}{c^2} \\ 0 \\ 0 \\ 0 \end{pmatrix} &
\begin{pmatrix} 0 \\ -\frac{\varepsilon\,\phi^{(1,0,0)}[x,y,z]}{c^2} \\ -\frac{\varepsilon\,\phi^{(0,1,0)}[x,y,z]}{c^2} \\ -\frac{\varepsilon\,\phi^{(0,0,1)}[x,y,z]}{c^2} \end{pmatrix} &
\begin{pmatrix} 0 \\ -\frac{\varepsilon\,\phi^{(0,1,0)}[x,y,z]}{c^2} \\ \frac{\varepsilon\,\phi^{(1,0,0)}[x,y,z]}{c^2} \\ 0 \end{pmatrix} &
\begin{pmatrix} 0 \\ -\frac{\varepsilon\,\phi^{(0,0,1)}[x,y,z]}{c^2} \\ 0 \\ \frac{\varepsilon\,\phi^{(1,0,0)}[x,y,z]}{c^2} \end{pmatrix} \\
\begin{pmatrix} \frac{\varepsilon\,\phi^{(0,1,0)}[x,y,z]}{c^2} \\ 0 \\ 0 \\ 0 \end{pmatrix} &
\begin{pmatrix} 0 \\ \frac{\varepsilon\,\phi^{(0,1,0)}[x,y,z]}{c^2} \\ -\frac{\varepsilon\,\phi^{(1,0,0)}[x,y,z]}{c^2} \\ 0 \end{pmatrix} &
\begin{pmatrix} 0 \\ -\frac{\varepsilon\,\phi^{(1,0,0)}[x,y,z]}{c^2} \\ -\frac{\varepsilon\,\phi^{(0,1,0)}[x,y,z]}{c^2} \\ -\frac{\varepsilon\,\phi^{(0,0,1)}[x,y,z]}{c^2} \end{pmatrix} &
\begin{pmatrix} 0 \\ 0 \\ -\frac{\varepsilon\,\phi^{(0,0,1)}[x,y,z]}{c^2} \\ \frac{\varepsilon\,\phi^{(0,1,0)}[x,y,z]}{c^2} \end{pmatrix} \\
\begin{pmatrix} \frac{\varepsilon\,\phi^{(0,0,1)}[x,y,z]}{c^2} \\ 0 \\ 0 \\ 0 \end{pmatrix} &
\begin{pmatrix} 0 \\ \frac{\varepsilon\,\phi^{(0,0,1)}[x,y,z]}{c^2} \\ 0 \\ -\frac{\varepsilon\,\phi^{(1,0,0)}[x,y,z]}{c^2} \end{pmatrix} &
\begin{pmatrix} 0 \\ 0 \\ \frac{\varepsilon\,\phi^{(0,0,1)}[x,y,z]}{c^2} \\ -\frac{\varepsilon\,\phi^{(0,1,0)}[x,y,z]}{c^2} \end{pmatrix} &
\begin{pmatrix} 0 \\ -\frac{\varepsilon\,\phi^{(1,0,0)}[x,y,z]}{c^2} \\ -\frac{\varepsilon\,\phi^{(0,1,0)}[x,y,z]}{c^2} \\ -\frac{\varepsilon\,\phi^{(0,0,1)}[x,y,z]}{c^2} \end{pmatrix}
\end{array}\right)^{\alpha}{}_{\beta\gamma}$$

Note : ϕ(1,0,0) means derivative of function ϕ(x,y,z) with respect to x. and so on..

Riemann Curvature Tensor:

In[30]:=
```
Priemann1 = Collect[Riem[-μ, -ν, -ρ, -σ] /. rules, ε, Simplify]
```

Out[30]=

$$\left(\begin{array}{ccc}
\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & \frac{\varepsilon\,\phi^{(2,0,0)}[x,y,z]}{c^2} & \frac{\varepsilon\,\phi^{(1,1,0)}[x,y,z]}{c^2} & \frac{\varepsilon\,\phi^{(1,0,1)}[x,y,z]}{c^2} \\ 0 & \frac{\varepsilon\,\phi^{(1,1,0)}[x,y,z]}{c^2} & \frac{\varepsilon\,\phi^{(0,2,0)}[x,y,z]}{c^2} & \frac{\varepsilon\,\phi^{(0,1,1)}[x,y,z]}{c^2} \\ 0 & \frac{\varepsilon\,\phi^{(1,0,1)}[x,y,z]}{c^2} & \frac{\varepsilon\,\phi^{(0,1,1)}[x,y,z]}{c^2} & \frac{\varepsilon\,\phi^{(0,0,2)}[x,y,z]}{c^2} \end{pmatrix} &
\begin{pmatrix} 0 & -\frac{\varepsilon\,\phi^{(2,0,0)}[x,y,z]}{c^2} & -\frac{\varepsilon\,\phi^{(1,1,0)}[x,y,z]}{c^2} & -\frac{\varepsilon\,\phi^{(1,0,1)}[x,y,z]}{c^2} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} & \cdots \\
\begin{pmatrix} 0 & 0 & 0 & 0 \\ -\frac{\varepsilon\,\phi^{(2,0,0)}[x,y,z]}{c^2} & 0 & 0 & 0 \\ -\frac{\varepsilon\,\phi^{(1,1,0)}[x,y,z]}{c^2} & 0 & 0 & 0 \\ -\frac{\varepsilon\,\phi^{(1,0,1)}[x,y,z]}{c^2} & 0 & 0 & 0 \end{pmatrix} &
\begin{pmatrix} \frac{\varepsilon\,\phi^{(2,0,0)}[x,y,z]}{c^2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\varepsilon\,(\phi^{(0,2,0)}[x,y,z]+\phi^{(2,0,0)}[x,y,z])}{c^2} & \frac{\varepsilon\,\phi^{(0,1,1)}[x,y,z]}{c^2} \\ 0 & 0 & \frac{\varepsilon\,\phi^{(0,1,1)}[x,y,z]}{c^2} & \frac{\varepsilon\,(\phi^{(0,0,2)}[x,y,z]+\phi^{(2,0,0)}[x,y,z])}{c^2} \end{pmatrix} & \cdots \\
\begin{pmatrix} 0 & 0 & 0 & 0 \\ -\frac{\varepsilon\,\phi^{(1,1,0)}[x,y,z]}{c^2} & 0 & 0 & 0 \\ -\frac{\varepsilon\,\phi^{(0,2,0)}[x,y,z]}{c^2} & 0 & 0 & 0 \\ -\frac{\varepsilon\,\phi^{(0,1,1)}[x,y,z]}{c^2} & 0 & 0 & 0 \end{pmatrix} &
\begin{pmatrix} \frac{\varepsilon\,\phi^{(1,1,0)}[x,y,z]}{c^2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -\frac{\varepsilon\,(\phi^{(0,2,0)}[x,y,z]+\phi^{(2,0,0)}[x,y,z])}{c^2} & 0 & -\frac{\varepsilon\,\phi^{(1,0,1)}[x,y,z]}{c^2} \\ 0 & -\frac{\varepsilon\,\phi^{(0,1,1)}[x,y,z]}{c^2} & 0 & \frac{\varepsilon\,\phi^{(1,1,0)}[x,y,z]}{c^2} \end{pmatrix} & \cdots \\
\begin{pmatrix} 0 & 0 & 0 & 0 \\ -\frac{\varepsilon\,\phi^{(1,0,1)}[x,y,z]}{c^2} & 0 & 0 & 0 \\ -\frac{\varepsilon\,\phi^{(0,1,1)}[x,y,z]}{c^2} & 0 & 0 & 0 \\ -\frac{\varepsilon\,\phi^{(0,0,2)}[x,y,z]}{c^2} & 0 & 0 & 0 \end{pmatrix} &
\begin{pmatrix} \frac{\varepsilon\,\phi^{(1,0,1)}[x,y,z]}{c^2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -\frac{\varepsilon\,\phi^{(0,1,1)}[x,y,z]}{c^2} & \frac{\varepsilon\,\phi^{(1,0,1)}[x,y,z]}{c^2} & 0 \\ 0 & -\frac{\varepsilon\,(\phi^{(0,0,2)}[x,y,z]+\phi^{(2,0,0)}[x,y,z])}{c^2} & -\frac{\varepsilon\,\phi^{(1,1,0)}[x,y,z]}{c^2} & 0 \end{pmatrix} & \cdots
\end{array}\right)$$

Ricci Tensor:

In[31]:=
```
Pricci1 = Collect[Ricci1[-μ, -ν] /. rules, ε, Simplify]
```

Out[31]=

$$\begin{pmatrix} \frac{\varepsilon\,(\phi^{(0,0,2)}[x,y,z]+\phi^{(0,2,0)}[x,y,z]+\phi^{(2,0,0)}[x,y,z])}{c^2} & 0 & 0 & \cdots \\ 0 & \frac{\varepsilon\,(\phi^{(0,0,2)}[x,y,z]+\phi^{(0,2,0)}[x,y,z]+\phi^{(2,0,0)}[x,y,z])}{c^2} & 0 & \cdots \\ 0 & 0 & \frac{\varepsilon\,(\phi^{(0,0,2)}[x,y,z]+\phi^{(0,2,0)}[x,y,z]+\phi^{(2,0,0)}[x,y,z])}{c^2} & \cdots \\ 0 & 0 & 0 & \varepsilon\,(\phi^{(0,0,2)}[x,y,z]+\phi^{(0,2,}\cdots \end{pmatrix}$$

Ricci Scalar:

In[32]:=
```
PRS1 = Collect[RS /. rules, ε, Simplify]
```

Out[32]=

$$\frac{2\,\varepsilon\,\left(\phi^{(0,0,2)}[x,y,z]+\phi^{(0,2,0)}[x,y,z]+\phi^{(2,0,0)}[x,y,z]\right)}{c^2}$$

I have read in the forum that , the developers of xAct are trying to make the transitioning between xPert and xCoba easier. hopefully it will be released soon which will eliminate the need to define the rules explicitly.

But right now we need to define the rules as done in the notebook.