In[∘]:=

```
Get@FileNameJoin@{NotebookDirectory[], "ParityOddSpinTwoScalar.m"};
```

Marco's parity-odd spin-two scalar

Connection to Part III project	The purpose of this script is to compute the scalar part of the parity–odd spin–two mode in the connection, by analogy to standard SVT techniques.

```
-------------------------------------------------------

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 linux executable...

Connection established.

-------------------------------------------------------

Package xAct`xTensor`  version 1.2.0, {2021, 10, 17}

CopyRight (C) 2002-2021, 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.

-------------------------------------------------------

-------------------------------------------------------

Package xAct`xPerm`  version 1.2.3, {2015, 8, 23}

CopyRight (C) 2003-2020, 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.

-------------------------------------------------------

Connecting to external linux executable...

Connection established.

-------------------------------------------------------

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.
```

```
** 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`Invar`  version 2.0.5, {2013, 7, 1}

CopyRight (C) 2006-2020, J. M. Martin-Garcia,
  D. Yllanes and R. Portugal, under the General Public License.

** DefConstantSymbol: Defining constant symbol sigma.

** DefConstantSymbol: Defining constant symbol dim.

** Option CurvatureRelations of DefCovD changed from True to False

** Variable $CommuteCovDsOnScalars changed from True to False

------------------------------------------------------

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.

------------------------------------------------------

Package xAct`SymManipulator`  version 0.9.5, {2021, 9, 14}

CopyRight (C) 2011-2021, Thomas Bäckdahl, under the General Public License.

------------------------------------------------------

Package xAct`xTras`  version 1.4.2, {2014, 10, 30}

CopyRight (C) 2012-2014, Teake Nutma, under the General Public License.

** Variable $CovDFormat changed from Postfix to Prefix

** Option CurvatureRelations of DefCovD changed from False to True

------------------------------------------------------

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.

------------------------------------------------------

------------------------------------------------------

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.

------------------------------------------------------------
```

This time define our manifold to be Euclidean, with three dimensions.

```
** DefManifold: Defining manifold M3.

** DefVBundle: Defining vbundle TangentM3.

** DefTensor: Defining symmetric metric tensor G[-a, -b].

** DefTensor: Defining antisymmetric tensor epsilonG[-a, -a1, -b].

** DefCovD: Defining covariant derivative CD[-a].

** DefTensor: Defining vanishing torsion tensor TorsionCD[a, -a1, -b].

** DefTensor: Defining symmetric Christoffel tensor ChristoffelCD[a, -a1, -b].

** DefTensor: Defining vanishing Riemann tensor RiemannCD[-a, -a1, -b, -b1].

** DefTensor: Defining vanishing Ricci tensor RicciCD[-a, -a1].

** DefTensor: Defining vanishing Ricci scalar RicciScalarCD[].

** DefTensor: Defining vanishing Einstein tensor EinsteinCD[-a, -a1].

** DefTensor: Defining vanishing Weyl tensor WeylCD[-a, -a1, -b, -b1].

** DefTensor: Defining vanishing TFRicci tensor TFRicciCD[-a, -a1].

** DefTensor: Defining vanishing Kretschmann scalar KretschmannCD[].

** DefTensor: Defining
 vanishing symmetrized Riemann tensor SymRiemannCD[-a, -a1, -b, -b1].

** DefTensor: Defining vanishing symmetric Schouten tensor SchoutenCD[-a, -a1].

** DefTensor: Defining symmetric cosmological Schouten tensor SchoutenCCCD[LI[_], -a, -a1].

** DefTensor: Defining symmetric cosmological Einstein tensor EinsteinCCCD[LI[_], -a, -a1].

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

Define a Planck mass.

```
** DefConstantSymbol: Defining constant symbol MPl.

** DefTensor: Defining tensor Phi[].
```

Now we define the rank-three tensor which will represent the mode.

```
** DefTensor: Defining tensor T[-a, -b, -c].
```

$$\mathcal{T}_{\alpha\beta\chi} \qquad (1)$$

Note that we don't assume any symmetry for this tensor yet. We will build an ansatz below, and gradually symmetrise it.

For example the antisymmetry of the Levi-Civita symbols means that there is no pseudoscalar mode buried in this tensor, since we know that the tensor will end up being symmetric in its first pair of indices. For now however, a contraction with the epsilon suggests that there might be such a mode.

$$\epsilon\delta_{\alpha\beta\chi}\ \epsilon\delta^{\delta\epsilon\phi}\ \mathcal{T}_{\delta\epsilon\phi} \tag{2}$$

$$\left(-\delta_{\alpha}{}^{\phi}\ \delta_{\beta}{}^{\epsilon}\ \delta_{\chi}{}^{\delta}+\delta_{\alpha}{}^{\epsilon}\ \delta_{\beta}{}^{\phi}\ \delta_{\chi}{}^{\delta}+\delta_{\alpha}{}^{\phi}\ \delta_{\beta}{}^{\delta}\ \delta_{\chi}{}^{\epsilon}-\delta_{\alpha}{}^{\delta}\ \delta_{\beta}{}^{\phi}\ \delta_{\chi}{}^{\epsilon}-\delta_{\alpha}{}^{\epsilon}\ \delta_{\beta}{}^{\delta}\ \delta_{\chi}{}^{\phi}+\delta_{\alpha}{}^{\delta}\ \delta_{\beta}{}^{\epsilon}\ \delta_{\chi}{}^{\phi}\right)\ \mathcal{T}_{\delta\epsilon\phi} \tag{3}$$

$$\mathcal{T}_{\alpha\beta\chi}-\mathcal{T}_{\alpha\chi\beta}-\mathcal{T}_{\beta\alpha\chi}+\mathcal{T}_{\beta\chi\alpha}+\mathcal{T}_{\chi\alpha\beta}-\mathcal{T}_{\chi\beta\alpha} \tag{4}$$

Another point is that I'd previously recommended you use some defined functions to construct your ansatz. We can try this, e.g. as follows.

$$\text{MakeContractionAnsatz}\left[\delta_{\alpha\beta}\ \partial_{\chi}\phi, \{-a, -b, -c\}\right] \tag{5}$$

Try it yourself by removing the comments, but as suspected, this is pretty trivial! As often happens in computer algebra, it is much easier to do the whole thing by hand.

Define first some constants.

```
** DefConstantSymbol: Defining constant symbol C1.

** DefConstantSymbol: Defining constant symbol C2.

** DefConstantSymbol: Defining constant symbol C3.

** DefConstantSymbol: Defining constant symbol C4.
```

Now develop the ansatz.

$$\mathcal{T}_{\alpha\beta\chi} \tag{6}$$

$$\text{C2}\ \delta_{\beta\chi}\ \partial_{\alpha}\phi+\text{C4}\ \epsilon\delta_{\beta\chi\alpha'}\ \partial^{\alpha'}\partial_{\alpha}\phi+\text{C4}\ \epsilon\delta_{\alpha\chi\alpha'}\ \partial^{\alpha'}\partial_{\beta}\phi+\text{C2}\ \delta_{\alpha\chi}\ \partial_{\beta}\phi+\text{C1}\ \delta_{\alpha\beta}\ \partial_{\chi}\phi+\text{C3}\,\partial_{\chi}\partial_{\beta}\partial_{\alpha}\phi \tag{7}$$

So that looks like a fairly general ansatz, we have four unknown coefficients.

The only restriction in concocting this ansatz that we imposed so far was for symmetry in the first pair of indices (since it was so easy to insist on). Let's just check that this symmetry works out.

$$\mathcal{T}_{\alpha\beta\chi}-\mathcal{T}_{\beta\alpha\chi}==0 \tag{8}$$

$$0==0 \tag{9}$$

Okay, so that is fine. The symmetry condition knocks out 3*3=9 d.o.f, and it needs no extra condition on the unknown constants to be already true. We have 3*3*3-9=18 d.o.f remaining.

Now let's try the no-trace condition across the non-symmetric indices, which knocks out 3 d.o.f. We have 18-3=15 d.o.f remaining.

$$\mathcal{T}_{\alpha\beta}{}^{\alpha} == 0 \tag{10}$$

$$\mathrm{C3}\,\partial_{\alpha}\partial_{\beta}\partial^{\alpha}\phi + (\mathrm{C1} + 4\,\mathrm{C2})\,\partial_{\beta}\phi == 0 \tag{11}$$

So from this we learn that we need the following constraints.

$$\mathrm{C1} + 4\,\mathrm{C2} == 0 \;\&\&\; \mathrm{C3} == 0 \tag{12}$$

Now we move on to the trace on the symmetric pair of indices. Again we have 15-3=12 d.o.f remaining.

$$\mathcal{T}_{\alpha}{}^{\alpha}{}_{\chi} == 0 \tag{13}$$

$$\mathrm{C3}\,\partial_{\alpha}\partial^{\alpha}\partial_{\chi}\phi + \mathrm{C4}\;\epsilon\delta_{\chi\alpha\alpha'}\;\partial^{\alpha'}\partial^{\alpha}\phi + (3\,\mathrm{C1} + 2\,\mathrm{C2})\,\partial_{\chi}\phi == 0 \tag{14}$$

So from this we learn that we need the following constraints.

$$3\,\mathrm{C1} + 2\,\mathrm{C2} == 0 \;\&\&\; \mathrm{C3} == 0 \;\&\&\; \mathrm{C4} == 0 \tag{15}$$

However, this is a problem since the final constraint in this series is spurious. To see this, notice how it comes from the term where the Levi-Civita symbol is contracted over a spatial Hessian operator. Of course this term must vanish, because div of curl vanishes! The reason xAct can't get rid of this term has to do with the fact that covariant derivatives are generally non-commuting. There are ways around this using xTras or perhaps PD, but for now it is just a minor bug. The important result is that the system imposed on C1 and C2 by this set of constraints and the previous set is singular: both C1 and C2 must be zero, along with C3, though C4 need not be.

Connection to Part III project	Now let's try that cyclic identity. Counting the number of d.o.f removed by this identity is a bit of a fiddle. To understand what is happening, I think you can argue as follows. Consider instances of the equation where all indices are unequal, there are 3∗2∗1=6 of these. However you can massage each such instance using the symmetry in the first pair of indices to show that they all remove only 1 d.o.f. Next consider cases where only two indices are equal, there are 3∗2+3∗2+3∗2=18 of these, but again 3∗2+3∗2=12 are doubly–counted when you use symmetry in the first pair, leaving apparently 6 d.o.f removed by such cases. Next there are 3∗1∗1=3 cases with all indices equal. Okay, so that means 1+6+3=10 d.o.f are additionally removed. But this is a disaster since we whittled the tensor down to 12 d.o.f by the time we considered the cyclic symmetry, so we would have only 2 d.o.f left to span a massive spin–2 mode, which ought to have 2∗2+1=5 d.o.f. What has happened, is that you can actually pair up each of the all–indices–equal equations with the appropriate two of the two–indices–equal equations to show that the first–pair–traceless condition actually implies the first–last–pair–traceless condition (or vice versa). Therefore, either the cyclic identity over–counts the number of removed d.o.f by three, or we over–counted by three before we got here, depending on perspective. Accordingly, the condition is indeed enough to leave us with a spin–2 state. You should certainly prove this convincingly in your Part III report, rather than relying on this commentary.

$$\mathcal{T}_{\alpha\beta\chi}+\mathcal{T}_{\beta\chi\alpha}+\mathcal{T}_{\chi\alpha\beta}==0 \tag{16}$$

$$(\text{C1}+2\,\text{C2})\;\delta_{\beta\chi}\;\partial_{\alpha}\phi+(\text{C1}+2\,\text{C2})\;\delta_{\alpha\chi}\;\partial_{\beta}\phi+(\text{C1}+2\,\text{C2})\;\delta_{\alpha\beta}\;\partial_{\chi}\phi+3\,\text{C3}\,\partial_{\chi}\partial_{\beta}\partial_{\alpha}\phi==0 \tag{17}$$

So from this we learn that we need the following constraints.

$$C1 + 2\,C2 == 0\ \&\&\ 3\,C3 == 0 \qquad (18)$$

That's fine, it is consistent with everything apart from C4 vanishing.

Finally, for pedantry, we confirm that the epsilon contraction goes away.

$$\epsilon\delta^{\alpha\beta\chi}\ \mathcal{T}_{\alpha\beta\chi} == 0 \qquad (19)$$

$$0 == 0 \qquad (20)$$

And it does!

Connection to Part III project	Based on this, it looks as though it should be very straightforward to knock up the scalar part of the parity–odd spin–2 mode.