

GraClo Module Documentation 01

Sam Close
s.close@lancaster.ac.uk

January 21, 2021

General Usage

This package is for finding the independent components, their number, and equations relating components of a given tensor, and corresponding equations and symmetries, in a certain dimension.

Installation

1. Download the `GraClo vN.mpl` file.
2. Place the file in the Maple directory you are working in.
3. Open Maple.
4. Read the package into Maple by using `> read("GraClo.mpl")`.
5. To use the package, as with every Maple package, then enter `> with(GraClo)`.
This will allow you to use the commands with no prefix.

There is an alternative to the final step: you can enter `GraClo[command](...)` or `GraClo:-command(...)`.

Ancillary Functions

SYM

SYM is a procedure for creating a sum of permutations (symmetrising), described mathematically by

$$\Psi_{x_1 x_2 \dots x_n (x_3 x_4 \dots x_N)} = \frac{1}{k!} \sum_{\substack{\text{all permutations} \\ i_1 \dots i_k}} \Psi_{x_1 x_2 \dots x_n (x_{i_1} x_{i_2} \dots x_{i_k})}$$

where in this case, Ψ and its associated indices is an arbitrary tensor. Equivalently, this is equally valid for contravariant tensor components.

SYM takes two arguments: **condition**, and **tenss**¹. To begin with the latter, **tenss** is the tensor over which you are performing the symmetry operation. **condition** is the (you guessed it), symmetry condition, input as a list. For example, suppose we want to find the symmetry of a, b, c , in the tensor Ψ_{abcd} :

```
> SYM([a,b,c],Psi[a,b,c,d]);
```

$$\frac{\Psi_{a,b,c,d}}{6} + \frac{\Psi_{a,c,b,d}}{6} + \frac{\Psi_{b,a,c,d}}{6} + \frac{\Psi_{b,c,a,d}}{6} + \frac{\Psi_{c,a,b,d}}{6} + \frac{\Psi_{c,b,a,d}}{6}$$

ASYM

Similar to the above, **ASYM** is a procedure for finding the anti-symmetric sum of permutations, described mathematically by

$$\Psi_{x_1 x_2 \dots x_n (x_3 x_4 \dots x_N)} = \frac{1}{k!} \sum_{\substack{\text{all permutations} \\ i_1 \dots i_k}} \epsilon_{i_1 i_2 \dots i_k} \Psi_{x_1 x_2 \dots x_n (x_{i_1} x_{i_2} \dots x_{i_k})}$$

where ϵ is the Levi-Civita Symbol, being 1 for even permutations, and -1 for odd permutations. Consider for this case, the Riemann Curvature tensor, R_{abcd} with the asymmetry on b, c, d

```
> ASYM([b,c,d],R[a,b,c,d])
```

$$\frac{R_{a,b,c,d}}{6} - \frac{R_{a,b,d,c}}{6} - \frac{R_{a,c,b,d}}{6} + \frac{R_{a,c,d,b}}{6} + \frac{R_{a,d,b,c}}{6} - \frac{R_{a,d,c,b}}{6}$$

As in **SYM**, the condition is input as a list.

Primary Functions

Each of the following functions take the same arguments: **tenss**, **basisEquations**, and **dim**, in that order. As in the **SYM** and **ASYM** commands, **tenss** is the tensor in question (e.g., $T^{\mu\nu} \rightarrow T[\text{mu}, \text{nu}]$). This is necessary currently to extract the rank, indices, and base. **basisEquations** takes a list of equations, written in free index form. For instance, using the Lanczos tensor's symmetries [1] as an example, they would be entered as a list as follows:

```
> [H[a,b,c] + H[b,a,c] = 0, H[a,b,c] + H[b,c,a] + H[c,a,b] = 0]
```

$$[H_{a,b,c} + H_{b,a,c} = 0, H_{a,b,c} + H_{b,c,a} + H_{c,a,b} = 0]$$

or equivalently, using the symmetry commands

```
> [SYM([a,b],H[a,b,c]) = 0, ASYM([a,b,c],H[a,b,c]) = 0].
```

¹It is called **tenss** because that is what I wrote at the time, so as not to confuse myself with other appearances of the word 'tensor'.

$$[\frac{H_{a,b,c}}{2} + \frac{H_{b,a,c}}{2} = 0, \frac{H_{a,b,c}}{6} - \frac{H_{a,c,b}}{6} - \frac{H_{b,a,c}}{6} + \frac{H_{b,c,a}}{6} + \frac{H_{c,a,b}}{6} - \frac{H_{c,b,a}}{6} = 0]$$

`dim` is, somewhat intuitively, the dimension of the system. By default, we consider the dimensions running from $(0, 1, \dots, \text{dim} - 1)$. If you wish to alter this, to each of the following commands you can use the option `startdim` to set the starting dimension (i.e., `startdim = 1` would mean the dimensions run from $(1, 2, \dots, \text{dim})$). Due to the limitations of `convert`, this means that we require $\text{dim} \geq 2$: though it should be obvious that if you are in one dimension, there is only one component.

If you do not input any equations into `basisEquations`, but simply enter `[]`, then you will be dealing with all the components of the tensor.

IndependentComponents

As the name would suggest, this solves for, and returns a list of the independent components of a given tensor, given the equations governing the symmetry, and the dimension. For a full worked example, let us consider finding the components of the Riemann Curvature tensor, in **five** dimensions.

```
> with(GraClo):
> eqns := [R[a,b,c,d] = -R[b,a,c,d], R[a,b,c,d] = -R[a,b,d,c],
R[a,b,c,d] = R[c,d,a,b], ASYM([b,c,d], R[a,b,c,d]) = 0];

eqns := [Ra,b,c,d = -Rb,a,c,d, Ra,b,c,d = -Ra,b,d,c, Ra,b,c,d = Rc,d,a,b,
 $\frac{R_{a,b,c,d}}{6} - \frac{R_{a,b,d,c}}{6} - \frac{R_{a,c,b,d}}{6} + \frac{R_{a,c,d,b}}{6} + \frac{R_{a,d,b,c}}{6} - \frac{R_{a,d,c,b}}{6} = 0]$ 

> IndependentComponents(R[a,b,c,d], eqns, 5);

[R0,2,1,0, R0,3,1,0, R0,3,2,0, R0,4,1,0, R0,4,2,0, R0,4,3,0, R1,0,0,1, R1,2,1,0, R1,3,1,0,
R1,3,2,1, R1,4,1,0, R1,4,2,1, R1,4,3,1, R2,0,0,2, R2,1,1,2, R2,1,2,0, R2,1,3,0, R2,1,4,0,
R2,3,2,0, R2,3,2,1, R2,4,2,0, R2,4,2,1, R2,4,3,2, R3,0,0,3, R3,1,1,3, R3,1,2,0, R3,1,3,0,
R3,1,4,0, R3,2,2,3, R3,2,3,0, R3,2,3,1, R3,2,4,0, R3,2,4,1, R3,4,3,0, R3,4,3,1, R3,4,3,2,
R4,0,0,4, R4,1,1,4, R4,1,2,0, R4,1,3,0, R4,1,4,0, R4,2,2,4, R4,2,3,0, R4,2,3,1, R4,2,4,0,
R4,2,4,1, R4,3,3,4, R4,3,4,0, R4,3,4,1, R4,3,4,2]
```

NumberOfIndependentComponents

As (again) the name suggests, this just returns the number of independent components, if it is not necessary to know which components they are. This is the equivalent of course, of just performing `nops` or `numelems` on the list of independent components. Continuing our code from above,

```
> NumberOfIndependentComponents(R[a,b,c,d], eqns, 5);
```

Gratefully, this is what we expect from the Riemann tensor in five dimensions. As proven algebraically in [2], there are $\frac{n^2}{12}(n^2 - 1)$ independent components in an n dimensional Riemann Curvature tensor.

Equations

This name is slightly more abstract, but it returns a list of the other components, in terms of the independent components. Since there are a large number of non-zero components, in this instance, I will reduce the dimension to 3, but keep the equations the same

```
> Equations(R[a,b,c,d],eqns,3);
```

$$[R_{0,1,0,1} = -R_{1,0,0,1}, R_{0,1,0,2} = -R_{0,2,1,0}, R_{0,1,1,0} = R_{1,0,0,1}, R_{0,1,1,2} = -R_{1,2,1,0},$$

$$R_{0,1,2,0} = R_{0,2,1,0}, R_{0,1,2,1} = R_{1,2,1,0}, R_{0,2,0,1} = -R_{0,2,1,0}, R_{0,2,0,2} = -R_{2,0,0,2},$$

$$R_{0,2,1,2} = R_{2,1,2,0}, R_{0,2,2,0} = R_{2,0,0,2}, R_{0,2,2,1} = -R_{2,1,2,0}, R_{1,0,0,2} = R_{0,2,1,0},$$

$$R_{1,0,1,0} = -R_{1,0,0,1}, R_{1,0,1,2} = R_{1,2,1,0}, R_{1,0,2,0} = -R_{0,2,1,0}, R_{1,0,2,1} = -R_{1,2,1,0},$$

$$R_{1,2,0,1} = -R_{1,2,1,0}, R_{1,2,0,2} = R_{2,1,2,0}, R_{1,2,1,2} = -R_{2,1,1,2}, R_{1,2,2,0} = -R_{2,1,2,0},$$

$$R_{1,2,2,1} = R_{2,1,1,2}, R_{2,0,0,1} = R_{0,2,1,0}, R_{2,0,1,0} = -R_{0,2,1,0}, R_{2,0,1,2} = -R_{2,1,2,0},$$

$$R_{2,0,2,0} = -R_{2,0,0,2}, R_{2,0,2,1} = R_{2,1,2,0}, R_{2,1,0,1} = R_{1,2,1,0}, R_{2,1,0,2} = -R_{2,1,2,0},$$

$$R_{2,1,1,0} = -R_{1,2,1,0}, R_{2,1,2,1} = -R_{2,1,1,2}]$$

In this instance, it may not have been that illustrative the usefulness of this equation, since there is only one term on each side.

Limitations

- Cannot work with ODEs or PDEs (e.g., $\partial_\mu \zeta^{\mu\nu}_{\rho\kappa} = 0$ or anything remotely like that)
- There is no way to work explicitly with covariant and contravariant indices.
- It cannot work with components explicitly stated as functions (e.g., $\gamma^{\mu\nu}(\sigma)$) as Maple seems to always try to solve for the independent variable, even when it is specified not to.

Why not use Physics tensors?

The **Physics** package does have many useful features, and is far more advanced in terms of tensorial capability. However, this package does have limitations for independent components, due to the limited scope of the **Library:-NumberOfIndependentTensorComponents** command and the way tensors are defined. Contrast using **Physics** and **Graclo** to find the number of independent components of the Riemann curvature tensor in four dimensions.

```

> with(Physics):
> Define(R[mu,nu,alpha,beta],symmetric={{[1,2],[3,4]}},
antisymmetric = {[1,2],[3,4]},minimizetensorcomponents)

$$\{\gamma_\mu, \sigma_\mu, R_{\alpha,\beta,\mu,\nu}, \partial_\mu, g_{\mu,\nu}, \epsilon_{\alpha,\beta,\mu,\nu}\}$$

> Library:-NumberOfIndependentTensorComponents(R[mu,nu,alpha,beta])

```

21

This, as discussed above, is not the correct number of independent components for the Riemann curvature tensor in four-dimensions. Importantly, it is down to the way **GraClo** takes conditions. **GraClo** can handle any number of algebraic relationships that describe the tensor, but **Physics** currently cannot.

Examples

Example 1: Riemann Curvature Tensor in Four-Dimensions

Granted this tensor has been done to death, but herein follows a fully worked example, finding the number of independent components of the four dimensional Riemann Curvature Tensor.

```

> read("GraClo vN.mpl"):
> with(GraClo):
[ASYM,Equations,IndependentComponents,NumberOfIndependentComponents,SYM]
> eqns := [R[a,b,c,d] = R[c,d,a,b], R[a,b,c,d] = -R[b,a,c,d],
R[a,b,c,d] = -R[a,b,d,c], ASYM([b,c,d],R[a,b,c,d]) = 0]

$$[R_{a,b,c,d} = R_{c,d,a,b}, R_{a,b,c,d} = -R_{b,a,c,d}, R_{a,b,c,d} = -R_{a,b,d,c},$$


$$\frac{R_{a,b,c,d}}{6} - \frac{R_{a,b,d,c}}{6} - \frac{R_{a,c,b,d}}{6} + \frac{R_{a,c,d,b}}{6} + \frac{R_{a,d,b,c}}{6} - \frac{R_{a,d,c,b}}{6} = 0]$$

> IndependentComponents(R[a,b,c,d],eqns,4,1)
[R1,3,2,1, R1,4,2,1, R1,4,3,1, R2,1,1,2, R2,3,2,1, R2,4,2,1, R2,4,3,2, R3,1,1,3, R3,2,2,3, R3,2,3,1,
R3,2,4,1, R3,4,3,1, R3,4,3,2, R4,1,1,4, R4,2,2,4, R4,2,3,1, R4,2,4,1, R4,3,3,4, R4,3,4,1, R4,3,4,2]

```

Example 2: A random combination of tensors

```

> read("GraClo vN.mpl"):
> with(GraClo):
> eqns := [SYM([i,j,k],P[i,j,k]) = L[i,j,k] - L[j,k,i],ASYM([i,j],L[i,j,k]) = 0]

$$[\frac{P_{i,j,k}}{6} + \frac{P_{i,k,j}}{6} + \frac{P_{j,i,k}}{6} + \frac{P_{j,k,i}}{6} + \frac{P_{k,i,j}}{6} + \frac{P_{k,j,i}}{6} = L_{i,j,k} - L_{j,k,i}, \frac{L_{i,j,k}}{2} - \frac{L_{j,i,k}}{2} = 0]$$

> [seq(NumberOfIndependentComponents(P[i,j,k],eqns,i),i=2..11)];
> [seq(NumberOfIndependentComponents(L[i,j,k],eqns,i),i=2..11)];
[4, 17, 44, 90, 160, 259, 392, 564, 780, 1045]
[2, 7, 16, 30, 50, 77, 112, 156, 210, 275]

```

References

- [1] Agacy RL. LETTER: A Note on the Algebraic Symmetries of the Riemann and Lanczos Tensors. General Relativity and Gravitation. 1999 Feb;31(2):219–222. Available from: <https://doi.org/10.1023/A:1018844227423>.
- [2] Bergmann PG. Introduction to the Theory of Relativity. New York; 1976.