

# GraClo Module Documentation

Sam Close  
[sam.w.close@gmail.com](mailto:sam.w.close@gmail.com)

April 23, 2022

## Contents

<b>1 Ancillary Procedures</b>	<b>1</b>
<b>2 Primary Procedures</b>	<b>2</b>
<b>3 Why not use Physics tensors for this?</b>	<b>4</b>
<b>4 Limitations</b>	<b>5</b>

## 1 Ancillary Procedures

These procedures are not a core part of the code, but are provided for making life easier when entering arguments for core procedures.

**SYM** SYM is a procedure for creating a sum of permutations (symmetrising) of an indexed object. For an indexed object  $T_{i_1 i_2 \dots i_n}$ , the symmetric operation is denoted with parentheses around the symmetric indices as

$$T_{(i_1 i_2 \dots i_n)} := \frac{1}{n!} \sum_{k=1}^n T_{\sigma_k(n)}$$

where  $\sigma_k(n)$  is the  $k$ th permutation of  $n$  indices. For instance, consider  $T_{i(jkl)}$ :

$$T_{i(jkl)} = \frac{1}{6} (T_{ijkl} + T_{iklj} + T_{iljk} + T_{ijlk} + T_{ikjl} + T_{ilkj}).$$

**SYM** takes two arguments: the indices over which to do the symmetrisation as a list, and the indexed object in question, and returns the sum (type ``+``). For example, considering the above object:

---

```
> SYM([j,k,l],T[i,j,k,l]);
```

$$\frac{T_{i,j,k,l}}{6} + \frac{T_{i,j,l,k}}{6} + \frac{T_{i,k,j,l}}{6} + \frac{T_{i,k,l,j}}{6} + \frac{T_{i,l,j,k}}{6} + \frac{T_{i,l,k,j}}{6}.$$

---

**ASYM** ASYM is a procedure for creating an alternating sum of permutations (antisymmetrising) of an indexed object. For an indexed object  $T_{i_1 i_2 \dots i_n}$ , the antisymmetric operation is denoted with square brackets around the indices as

$$T_{[i_1 i_2 \dots i_n]} := \frac{1}{n!} \sum_{k=1}^n \text{sign}(\sigma_k(n)) T_{\sigma_k(n)}$$

where  $\sigma_k(n)$  is the  $k$ th permutation of  $n$  indices, and sign means the parity of that permutation. For instance, consider  $T_{i[jkl]}$ :

$$T_{i[jkl]} = \frac{1}{6} (T_{ijkl} + T_{iklj} + T_{iljk} - T_{ijlk} - T_{ikjl} - T_{ilkj}).$$

ASYM takes two arguments: the indices over which to do the antisymmetrisation as a list, and the indexed object in question, and returns the alternating sum (type ``+``).

---

```
> ASYM([j,k,l],T[i,j,k,l]);
```

$$\frac{T_{i,j,k,l}}{6} - \frac{T_{i,j,l,k}}{6} - \frac{T_{i,k,j,l}}{6} + \frac{T_{i,k,l,j}}{6} + \frac{T_{i,l,j,k}}{6} - \frac{T_{i,l,k,j}}{6}.$$


---

## 2 Primary Procedures

**IndependentComponents** **IndependentComponents** is a procedure for finding the independent components of a given indexed object, in a particular dimension with accompanying equations. It takes one required positional argument: the indexed object; one optional positional argument: the dimension (default: 4); and two optional arguments: equations (default: NULL) and start dimension (default: 0). It returns a list.

Independent components are components with which one can represent all other components. For instance, in an easy example, consider the metric tensor  $g_{\mu\nu}$  in four-dimensions. It is symmetric, and it is well known to have 10 independent components.

---

```
> IndependentComponents(g[mu,nu],equations=[g[mu,nu]=g[nu,mu]]);
```

$$[g_{0,0}, g_{1,0}, g_{1,1}, g_{2,0}, g_{2,1}, g_{2,2}, g_{3,0}, g_{3,1}, g_{3,2}, g_{3,3}]$$

---

The choice of which is independent in this case is somewhat arbitrary, but Maple chooses this order. If one wished to work from  $(1-4)$  rather than  $(0-3)$ :

---

```
> IndependentComponents(g[mu,nu],equations=[g[mu,nu]=g[nu,mu]],startdim=1);
```

$$[g_{1,1}, g_{2,1}, g_{2,2}, g_{3,1}, g_{3,2}, g_{3,3}, g_{4,1}, g_{4,2}, g_{4,3}, g_{4,4}]$$


---

**NumberOfIndependentComponents** **NumberOfIndependentComponents** is a procedure for finding the number of independent components of a given indexed object, in a particular dimension with accompanying equations. It takes one required positional argument: the indexed object; one optional positional argument: the dimension (default: 4); and two optional arguments: equations (default: NULL) and start dimension (default: 0). It returns an integer.

As an example, consider the Christoffel symbols  $\Gamma_{\rho\mu\nu}$  which are symmetric in their last two indices, in four-dimensions.

---

```
> NumberOfIndependentComponents(Gamma[rho,mu,nu],equations=[Gamma[rho,mu,nu]=Gamma[rho,nu,mu]])
```

40

---

**ComponentRelationships** **ComponentRelationships** is a procedure for finding the equations relating components of a given indexed object in terms of the object's independent components, in a particular dimension with accompanying equations. It takes one required positional argument: the indexed object; one optional positional argument: the dimension (default: 4); and two optional arguments: equations (default: NULL) and start dimension (default: 0). It returns a list.

As an example, consider the Faraday tensor  $F_{\mu\nu}$  which is antisymmetric in its indices:

---

```
> ComponentRelationships(F[mu,nu],4,equations=[F[mu,nu]=-F[nu,mu]])
```

$$[F_{0,0} = 0, F_{0,1} = -F_{1,0}, F_{0,2} = -F_{2,0}, F_{0,3} = -F_{3,0}, \\ F_{1,1} = 0, F_{1,2} = -F_{2,1}, F_{1,3} = -F_{3,1}, F_{2,2} = 0, F_{2,3} = -F_{3,2}, F_{3,3} = 0]$$


---

### 3 Why not use Physics tensors for this?

The **Physics** package in Maple is extensive and very powerful, however, this implementation has a few advantages.

1. Smaller codebase, so easier to debug. In total the code is at most 135-lines long. It also only uses minimal dependencies of **combinat**, **ArrayTools**, and **GroupTheory**.
2. Does not require the setup steps of **Physics**.
3. It recovers results **Physics** cannot.

In four dimensions, the Riemann tensor  $R_{\mu\nu\rho\sigma}$  has 20 independent components, generated by the symmetries:

$$\begin{aligned} R_{\mu\nu\rho\sigma} &= -R_{\mu\nu\sigma\rho} \quad \text{and} \quad R_{\mu\nu\rho\sigma} = -R_{\nu\mu\rho\sigma} \\ R_{\mu\nu\rho\sigma} &= R_{\rho\sigma\mu\nu} \\ R_{\mu\nu\rho\sigma} + R_{\mu\rho\sigma\nu} + R_{\mu\sigma\nu\rho} &= 0 \end{aligned}$$

To get this number, in **Graclo**:

---

```
> AntiSym := R[mu,nu,rho,sigma] = -R[nu,mu,rho,sigma], R[mu,nu,rho,sigma] = -
R[mu,nu,sigma,rho]
```

$$AntiSym := R_{\mu,\nu,\rho,\sigma} = -R_{\nu,\mu,\rho,\sigma}, R_{\mu,\nu,\rho,\sigma} = -R_{\mu,\nu,\sigma,\rho}$$

```
> SymSwap := R[mu,nu,rho,sigma] = R[rho,sigma,mu,nu]
```

$$SymSwap := R_{\mu,\nu,\rho,\sigma} = R_{\rho,\sigma,\mu,\nu}$$

```
> SymSum := R[mu,nu,rho,sigma] + R[mu,rho,sigma,nu] + R[mu,sigma,nu,rho] = 0
```

$$SymSum := R_{\mu,\nu,\rho,\sigma} + R_{\mu,\rho,\sigma,\nu} + R_{\mu,\sigma,\nu,\rho} = 0$$

```
> NumberOfIndependentComponents(R[mu,nu,rho,sigma],equations=[AntiSym,SymSwap,SymSum])
```

20

---

One cannot get 20 with **Physics** according to the [Maple documentation](#).

---

```
> Define(R[mu,nu,rho,sigma],symmetric=[[1,2],[3,4]],antisymmetric=[[1,2],[3,4]],
minimizetensorcomponents):
> Library:-NumberOfIndependentTensorComponents(R[mu,nu,rho,sigma])
```

---

This is not to say one shouldn't use `Physics`: indeed it is one of the most expansive packages available, and for doing any tensor calculus it is a clear recommendation. However, purely for finding relationships in symmetries, `GraClo` offers some benefit.

## 4 Limitations

`GraClo` has a couple of limitations, some of which may not be surmountable.

1. It does not support contravariant indices. This is not an issue for finding independent components, but it would be nice to have.
2. `NumberOfIndependentComponents` scales poorly with both dimension and rank. Ideally, it would calculate the number of components just from the indices given, however at present it has to solve for all independent components via call to `IndependentComponents` and count the number of elements.