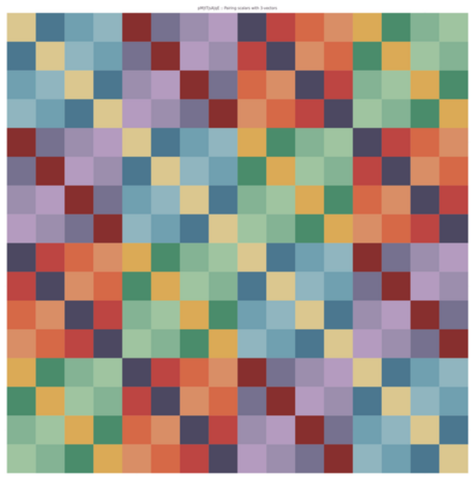
Innes D. Anderson-Morrison 2017

$\mathcal{D}_{\mathbb{S}}$: the 4Set differential



The 16 components of the algebra can be grouped into what I am calling *4-sets* that consist of a 3-vector **{x,y,z}** triplet ($\mathbb{B}$, $\mathbb{T}$, $\mathbb{A}$ and $\mathbb{E}$) paired with an additional element: $\boldsymbol{\alpha_p}$, $\boldsymbol{\alpha_0}$, $\boldsymbol{\alpha_{123}}$ and $\boldsymbol{\alpha_{0123}}$ respectively. From here on I will be referring to these four elements as **p**, **t**, **s** and **q** and the elements of each 3-vector as subscripts on their respective vector name: $\boldsymbol{\alpha_{23}}$ as $\mathbf{B_x}$ for example, with the paired additional element with the subscript **b** (for paired blade).

Under this notation $\mathbb{B} = \{\mathbf{B_b}, \mathbf{B_x}, \mathbf{B_y}, \mathbf{B_z}\} = \{\boldsymbol{\alpha_p}, \boldsymbol{\alpha_{23}}, \boldsymbol{\alpha_{31}}, \boldsymbol{\alpha_{12}}\}$ and the general 4-set is denoted $\mathbb{S} = \{\mathbf{S_b}, \mathbf{S_x}, \mathbf{S_y}, \mathbf{S_z}\}$ in boldface with **S** denoting the usual 3-vector elements.

A generalised 4-set differential can be defined for Cartesian coordinate as follows:

$$\mathcal{D}_{\mathbb{S}} = \partial/\alpha_\mu \partial_{\mathbb{S}\mu} \quad \forall\ \mu \in \{b, x, y, z\}$$

Where the sign of each term is determined by the metric in use.

In [1]:
```python
import arpy
from arpy import *

print('\narpy version: ', arpy.__version__)
print('\nAllowed αs: ', ', '.join([str(Alpha(a)) for a in ALLOWED]))
print('\nDivision: ', DIVISION_TYPE)
print('\nMetric: ', ''.join(['+' if i == 1 else '-' for i in METRIC]))
```

```
arpy version:  0.1.2

Allowed αs:  αp, α23, α31, α12, α0, α023, α031, α012, α123, α1, α2, α3, α0123, α10, α20, α30

Division:  into

Metric:  +---
```

Defining and using the operators within arpy

As covered in the docs, the way to define a new operator fitting the pattern described above is to use the **differential_operator** function. The list of α indices provided acts as the set of values that μ is allowed to take in the above definition.

Here we define the four 4-set differentials: ∇_{B}, $\mathcal{D}_T$, $\mathcal{D}_A$ and $\mathcal{D}_E$.

In [2]:
```python
DB = differential_operator(['p', '23', '31', '12'])
DT = differential_operator(['0', '023', '031', '012'])
DA = differential_operator(['123', '1', '2', '3'])
DE = differential_operator(['0123', '10', '20', '30'])
```

The action of such a differential is to always produce *16 components* that can be grouped into *5 vector calculus elements*.

In [3]:
```python
DB(XiT)
```

Out[3]:
```
{
  α0    (∂ₚξ₀, ∂₂₃ξ₀₂₃, ∂₃₁ξ₀₃₁, ∂₁₂ξ₀₁₂)
  α023  (-∂₂₃ξ₀, ∂ₚξ₀₂₃, ∂₁₂ξ₀₃₁, -∂₃₁ξ₀₁₂)
  α031  (-∂₃₁ξ₀, -∂₁₂ξ₀₂₃, ∂ₚξ₀₃₁, ∂₂₃ξ₀₁₂)
  α012  (-∂₁₂ξ₀, ∂₃₁ξ₀₂₃, -∂₂₃ξ₀₃₁, ∂ₚξ₀₁₂)
}
```

Note that when grouping into vector calculus del notation (https://en.wikipedia.org/wiki/Del) I am using the symbol $\boldsymbol{\nabla^S}$ to denote the standard 3-vector $\boldsymbol{\nabla}$ operator with **{x, y, z}** components being drawn from the 4-set $\mathbb{S}$.

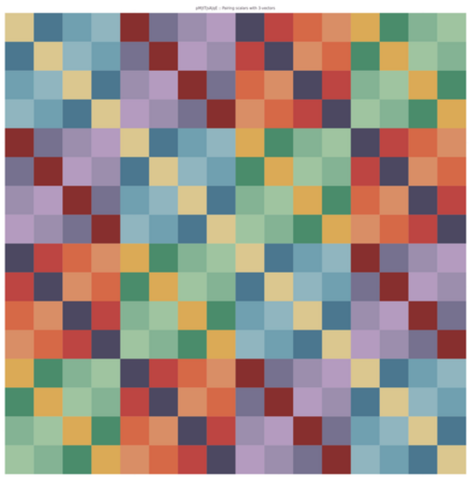
For $\mathcal{D}_{\mathbb{S}}\mathbb{S}'$ we obtain: $\partial_b \mathbb{S}_b'$ $\quad \partial_b \mathbb{S}'$ $\quad \nabla^S \cdot \mathbb{S}_v'$ $\quad \nabla^S\ \mathbb{S}'_b$ $\quad \nabla^S \times \mathbb{S}_v'$

In [4]:
```python
DT(XiA, as_del=True)
```

Out[4]:
```
{
  α0123 (-∇ᵀ•A, ∂₀ξ₁₂₃)
  αi0   (-∂₀A, -∇ᵀΞ₁₂₃, ∇ᵀxA)
}
```

Loading [Contrib]/a11y/accessibility-menu.js

The 5 ∂_b and ∇^S terms



This diagram (also shown above) is a visualisation of the Cayley table for the algebra that does not take sign into account. As such, it applies to all product based operations between αs and by extenstion, between 4-sets.

The colour of each cell denotes the resultant α value of the composition of the row α with the column α with αs in **B, T, A, E** order: **αp, α23, α31, α12, α0, α023, α031, α012, α123, α1, α2, α3, α0123, α10, α20, α30**.

When we compose one full 4-set with another we *always* obtain a result that is shown in the diagram as a single colour 4x4 block with tone denoting **{b, x, y, z}**.

As an example below, we take the full product of the MultiVectors **A** and **T** (red and purple respectively in the diagram) and obtain αs from **E** (green in the diagram).

In [5]:
```
ar('XiA ^ XiT')
```

Out[5]:
```
{
  α0123 (-ξ₁₂₃ξ₀, -ξ₁ξ₀₂₃, -ξ₂ξ₀₃₁, -ξ₃ξ₀₁₂)
  α10   (-ξ₁₂₃ξ₀₂₃, ξ₁ξ₀, ξ₂ξ₀₁₂, -ξ₃ξ₀₃₁)
  α20   (-ξ₁₂₃ξ₀₃₁, -ξ₁ξ₀₁₂, ξ₂ξ₀, ξ₃ξ₀₂₃)
  α30   (-ξ₁₂₃ξ₀₁₂, ξ₁ξ₀₃₁, -ξ₂ξ₀₂₃, ξ₃ξ₀)
}
```

In fact, the four 4-sets form a group under *full-product-like* composition (identifying divison as a full-product of one term with the inverse of another) with the Magnetic 4-set 𝔹 as the identity:

```
x | B T A E
-----------
B | B T A E
T | T B E A
A | A E B T
E | E A T B
```

The group formed is isomorphic to the Klein four-group (https://en.wikipedia.org/wiki/Klein_four-group) which has the group presentation (https://en.wikipedia.org/wiki/Presentation_of_a_group): **$< a, b \mid a^2 = b^2 = (ab)^2 = I >$**

(It should also be noted that the internal structor of each 4-set is also isomorphic to the Klein four-group.)

The general sign distribution of the algebra of course changes with respect to the operation being used, the metric and the choice of ordering for the elements of grade 2 and above. In practice, only the components of the electric field may be varied between **i0** and **0i**.

The following table shows the sign of each resulting component under full-product.

full product i0 +---

In [6]:
```
sign_cayley(full)
```

```
              B           A           T           E
         +---------+---------+---------+---------+
αp       | □ □ □ □ | □ □ □ □ | □ □ □ □ | □ □ □ □ |
α23      | □ ■ □ ■ | □ ■ □ ■ | ■ □ □ ■ | □ ■ □ ■ |
α31      | □ ■ ■ □ | □ ■ ■ □ | ■ ■ □ □ | □ ■ ■ □ |
α12      | □ □ ■ ■ | □ □ ■ ■ | ■ □ ■ □ | □ □ ■ ■ |
         +---------+---------+---------+---------+
α0       | □ □ □ □ | □ □ □ □ | □ ■ ■ ■ | □ ■ ■ ■ |
α023     | □ ■ □ ■ | □ ■ □ ■ | □ □ ■ □ | ■ ■ ■ □ |
α031     | □ ■ ■ □ | □ ■ ■ □ | □ □ □ ■ | ■ □ ■ ■ |
α012     | □ □ ■ ■ | □ □ ■ ■ | □ ■ □ □ | ■ ■ □ ■ |
         +---------+---------+---------+---------+
α123     | □ ■ ■ ■ | ■ ■ ■ ■ | □ ■ ■ ■ | ■ ■ ■ ■ |
α1       | □ □ □ ■ | □ ■ □ ■ | ■ ■ □ ■ | □ ■ □ ■ |
α2       | □ ■ □ □ | □ ■ ■ □ | ■ ■ ■ □ | □ ■ ■ □ |
α3       | □ □ ■ □ | □ □ ■ ■ | ■ □ ■ ■ | □ □ ■ ■ |
         +---------+---------+---------+---------+
α0123    | □ □ □ □ | ■ □ □ □ | □ ■ ■ ■ | ■ ■ ■ ■ |
α10      | □ ■ □ ■ | □ □ □ ■ | □ □ ■ □ | ■ □ ■ □ |
α20      | □ ■ ■ □ | □ ■ □ □ | □ □ □ ■ | ■ □ □ ■ |
α30      | □ □ ■ ■ | □ □ ■ □ | □ ■ □ □ | ■ ■ □ □ |
         +---------+---------+---------+---------+
```

Alternatively, the **sign_distribution** function can be used to look at what happens to each of the 5 groups of components under different setups. This also allows for specification of different metrics and allowed αs.

full product i0 -+++

In [7]:
```
sign_distribution(full, metric=(-1, 1, 1, 1))
```

```
∂b |□ □ □ □|   ∂Ξ |□ □ □ □|    ∇ |□ □ ■ □|   ∇• |■ ■ □ ■|   ∇x |■ ■ ■ ■|
   |□ ■ □ ■|      |□ ■ ■ □|      |□ ■ □ □|      |■ □ □ □|      |■ □ □ ■|
   |□ ■ ■ □|      |■ ■ □ □|      |□ □ □ ■|      |□ ■ □ □|      |■ ■ □ □|
   |□ □ ■ ■|      |□ ■ □ ■|      |□ ■ ■ ■|      |■ ■ ■ □|      |■ □ ■ □|
```

full product 0i +---

In [8]:
```
oi = ['p', '23', '31', '12', '0', '023', '031', '012', '123', '1', '2', '3', '0123', '01', '02', '03']
sign_distribution(full, allowed=oi)
```

```
∂b |□ □ □ □|   ∂Ξ |□ □ □ □|    ∇ |□ □ ■ ■|   ∇• |■ ■ □ □|   ∇x |□ □ □ □|
   |□ □ □ □|      |□ □ □ □|      |□ □ ■ ■|      |■ ■ □ □|      |□ □ □ □|
   |□ ■ □ ■|      |■ □ ■ □|      |□ ■ ■ □|      |□ ■ ■ □|      |□ ■ □ ■|
   |□ ■ □ ■|      |■ □ ■ □|      |□ ■ ■ □|      |□ ■ ■ □|      |□ ■ □ ■|
```

Generalised Force equations: $\mathbb{S}\mathcal{D}_{\mathbb{S}}\mathbb{S}$

As discussed in Dr Williamson's paper (http://eprints.gla.ac.uk/110966/), the generalised Lorentz Force (https://en.wikipedia.org/wiki/Lorentz_force) *may* be obtained by computing the full-product of F (the even subgroup of the $\mathbb{G}$ which is equal to B + E) with its space-time derivative $\mathcal{D}_\mu$F:

Force = F$\mathcal{D}_\mu$F

I say *may* as we are currently unsure as to whether this is the correct equation (which is the entire point of this notebook!) so we are looking into alternatives such as $\mathcal{D}_\mu$FF, $\mathcal{D}_\mu$(FF†), $\mathcal{D}_\mu(\mathbb{G}\mathbb{G}^\dagger)$ etc. I feel that it may be worthwhile looking at a generalisation of **F$\mathcal{D}_\mu$F** namely **$\mathbb{S}\mathcal{D}_{\mathbb{S}}\mathbb{S}$** where each of the three **$\mathbb{S}$** terms may be any 4-set.

The simplest case to consider is that of $\mathbb{B}\mathcal{D}_{\mathbb{B}}\mathbb{B}$

As $\mathbb{B}$ is the group identity it is self contained under composition with all elements of $\mathcal{D}_{\mathbb{B}}\mathbb{B}$ remaining within $\mathbb{B}$ and all elements of the product $\mathbb{B}\mathcal{D}_{\mathbb{B}}\mathbb{B}$ also remaining within $\mathbb{B}$. (*This can be seen from the leading diagonal of the Cayley table shown before: the composition of any 4-set with itself results in $\mathbb{B}$-terms.*)

Below I show the result of computing this product along with a partial simplification that is possible under the current version of arpy:

In [9]:
```
DBB = DB(XiB)
DB(XiB, as_del=True)
```

Out[9]:
```
{
  αp    (∇ᴮ•B, ∂ₚξₚ)
  αjk   (∂ₚB, -∇ᴮΞₚ, -∇ᴮxB)
}
```

In [10]:
```
BDBB = ar('XiB ^ DBB')
print('{} terms:'.format(len([term for term in BDBB])))
```

```
64 terms:
```

Loading [Contrib]/a11y/accessibility-menu.js

```
αₚ:
    ξₚ ( ∂ₚξₚ + ∂₂₃ξ₂₃ + ∂₃₁ξ₃₁ + ∂₁₂ξ₁₂ )
    + ξ₂₃( ∂₂₃ξₚ - ∂ₚξ₂₃ - ∂₁₂ξ₃₁ + ∂₃₁ξ₁₂ )
    + ξ₃₁( ∂₃₁ξₚ + ∂₁₂ξ₂₃ - ∂ₚξ₃₁ - ∂₂₃ξ₁₂ )
    + ξ₁₂( ∂₁₂ξₚ - ∂₃₁ξ₂₃ + ∂₂₃ξ₃₁ - ∂ₚξ₁₂ )

α₂₃:
    ξₚ( ∂₂₃ξₚ + ∂ₚξ₂₃ + ∂₁₂ξ₃₁ + ∂₃₁ξ₁₂ )
    + ξ₂₃( ∂ₚξₚ + ∂₂₃ξ₂₃ + ∂₃₁ξ₃₁ + ∂₁₂ξ₁₂ )
    + ξ₃₁( ∂₁₂ξₚ + ∂₃₁ξ₂₃ + ∂₂₃ξ₃₁ + ∂ₚξ₁₂ )
    + ξ₁₂( ∂₃₁ξₚ + ∂₁₂ξ₂₃ - ∂ₚξ₃₁ - ∂₂₃ξ₁₂ )

α₃₁:
    ξₚ( ∂₃₁ξₚ + ∂₁₂ξ₂₃ + ∂ₚξ₃₁ + ∂₂₃ξ₁₂ )
    + ξ₂₃( ∂₁₂ξₚ - ∂₃₁ξ₂₃ + ∂₂₃ξ₃₁ - ∂ₚξ₁₂ )
    + ξ₃₁( ∂ₚξₚ + ∂₂₃ξ₂₃ + ∂₃₁ξ₃₁ + ∂₁₂ξ₁₂ )
    + ξ₁₂( ∂₂₃ξₚ + ∂ₚξ₂₃ + ∂₁₂ξ₃₁ + ∂₃₁ξ₁₂ )

α₁₂:
    ξₚ( ∂₁₂ξₚ + ∂₃₁ξ₂₃ + ∂₂₃ξ₃₁ + ∂ₚξ₁₂ )
    + ξ₂₃( ∂₃₁ξₚ + ∂₁₂ξ₂₃ + ∂ₚξ₃₁ + ∂₂₃ξ₁₂ )
    + ξ₃₁( ∂₂₃ξₚ - ∂ₚξ₂₃ - ∂₁₂ξ₃₁ + ∂₃₁ξ₁₂ )
    + ξ₁₂( ∂ₚξₚ + ∂₂₃ξ₂₃ + ∂₃₁ξ₃₁ + ∂₁₂ξ₁₂ )
```

Or, in *partially* simplified notation we have this:

Note: further automated simplification is planned but this requires simplification of product terms accross common factors in addition to settling on the notation to use for some of the new grouped elements that have been identified. (See the next section for more details)

In [11]:
```
BDBB.simplified()
```

```
{
  αjk:
      ξₚ[∂ₚB, -∇ᴮΞₚ, -∇ᴮxB]
  α23:
      ξ₂₃[∇ᴮ•B, ∂ₚξₚ, -∂₁₂ξₚ, ∂₃₁ξ₂₃, -∂₂₃ξ₃₁, ∂ₚξ₁₂, -∂₃₁ξₚ, -∂₁₂ξ₂₃, ∂ₚξ₃₁, ∂₂₃ξ₁₂]
  α31:
      ξ₃₁[∇ᴮ•B, -∂₁₂ξₚ, ∂₃₁ξ₂₃, -∂₂₃ξ₃₁, ∂ₚξ₁₂, ∂ₚξₚ, -∂₂₃ξₚ, ∂ₚξ₂₃, ∂₁₂ξ₃₁, -∂₃₁ξ₁₂]
  α12:
      ξ₁₂[∇ᴮ•B, -∂₃₁ξₚ, -∂₁₂ξ₂₃, ∂ₚξ₃₁, ∂₂₃ξ₁₂, -∂₂₃ξₚ, ∂ₚξ₂₃, ∂₁₂ξ₃₁, -∂₃₁ξ₁₂, ∂ₚξₚ]
  αp:
      ξₚ[∇ᴮ•B, ∂ₚξₚ]
      ξ₂₃[-∂₂₃ξₚ, ∂ₚξ₂₃, ∂₁₂ξ₃₁, -∂₃₁ξ₁₂]
      ξ₃₁[-∂₃₁ξₚ, -∂₁₂ξ₂₃, ∂ₚξ₃₁, ∂₂₃ξ₁₂]
      ξ₁₂[-∂₁₂ξₚ, ∂₃₁ξ₂₃, -∂₂₃ξ₃₁, ∂ₚξ₁₂]
}
```

Simplifying further by hand I obtain the following:

Note: in the following expressions μνλ run over $\mathbb{B}_{x,y,z}$ (α_{23}, α_{31}, α_{12}) in cyclic left to right order . There are some terms using this notation that I think may be grouped onto α_{jk} in a similar way to Curl.

$\mathbb{B}\mathcal{D}_{\mathbb{B}}\mathbb{B}$ =

{

α_p: P[$\partial_p P + \nabla^B B$] + B[$\partial_{jk} P - B \bullet \partial_p B + B_\mu(\partial_\lambda B_\mu - \partial_\nu B_\lambda)$]

α_{jk}: P[$\partial_p B + \nabla^B P$] + $P.\partial_p P + B \bullet \nabla^B B$

α_μ: $P(\partial_{jk\lambda} B_\nu + \partial_{jk\nu} B_\lambda) + B_\lambda \partial_\nu P + B_\nu \partial_\nu B_\mu + (B_\nu \partial_\mu B_\nu - B_\lambda \partial_\mu B_\lambda) + (B_\nu \partial_p B_\lambda - B_\lambda \partial_p B_\nu)$

}

Each of the α_μ terms could be included in the α_{jk} group if we can come up with a name for what that operation *means*. I'm not 100% confident that each of those groups will ammount to a Physical description of something (in the way that Div, Grad and Curl do) but it may be worth further investigation. At the very least, it allows for a more compact representation of the result.

It is easy to see that we have a similar distribution of grouped elements (four 6-component Curl like terms for example) but that the original grouped term structure is not preserved. I think that this collection of terms can be generalised to the $\mathbb{S}\mathcal{D}_{\mathbb{S}}\mathbb{S}$ case modulo sign which will depend on the metric in use.