# Straining with dislocations, disclinations and lattice orientations:

TBB, after discussions with TR + CF, and document from CF – June 14 2013

Start with:

Where: = the elastic rotation tensor, X = axis, and = skew symmetric (or anti-symmetric) part of the elastic distortion tensor, :

Therefore:

Where is the Levi-Citta permutation symbol:

=1, when the order of the value for the subscripts is ijk, jki, kij

=-1, when the order of the value for the subscripts is ijk, jki, kji

=0, when the value of two (or three) subscripts are equal

And = 1 for i=j, and 0 otherwise.

Therefore (note these are sums of many elements, written out in ‘matrix like’ shape to make it clear where each part comes from):

For k = 1 (resolve permutation):

By inspection:

Note here that also:

is skew symmetric, therefore:

We are interested in curvature, , i.e. the spatial ‘gradient’ of this elastic rotation vector:

Evaluating this to give the tensor:

**In my notations, this is the transpose of the curvature tensor.**

Now build the curl of the tensor, disclination density, from:

Where = change in the elastic curvature, along the pth direction (I was just checking that the notation worked as I expect) as we note that:

Therefore:

i.e.

Therefore the components of the disclination density tensor are:

I redid the calculations below and agreed on these two relations. For the first one, with:  and . For example:  and . Therefore and



Therefore:

**Again, I consider this as the transpose of the disclination density tensor (but your derivations are correct).**

This seems to be the same as the pretty result that Claude shows.

1. Need to extend from just (mis-)orientation data to (mis-)orientation + strain

**There is no difference regarding the disclination density tensor: in contrast with the dislocation density tensor, it does not depend on the state of strain.**

1. How to ‘correct’ formation of the Nye tensor if we find out that the disclination density !=0

**No difference actually. You can use the same formula, i.e.,  (in component form: ) with the elastic curvature tensor as a gradient curl-free tensor (if the disclination tensor is zero) or not.**

1. How to consider the best ‘gradient’ functions, finite differences?

**I would say stupidly: you start with the standard one**

****

**)**

**and change it if it does not work? We could compare with Benoît’s method on the same map to check for possible differences.**