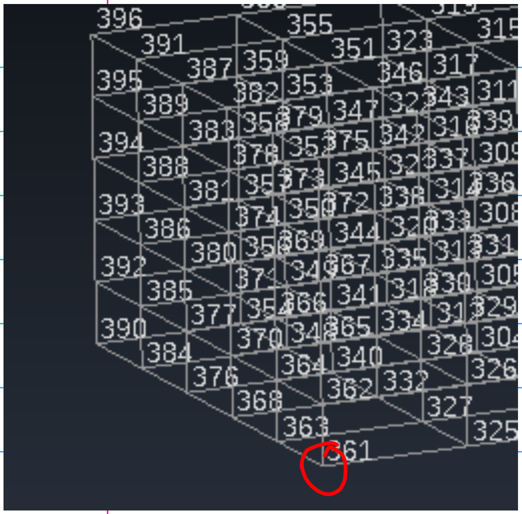
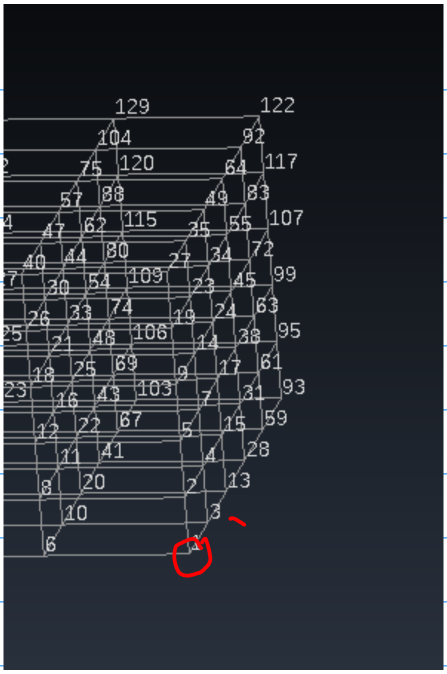
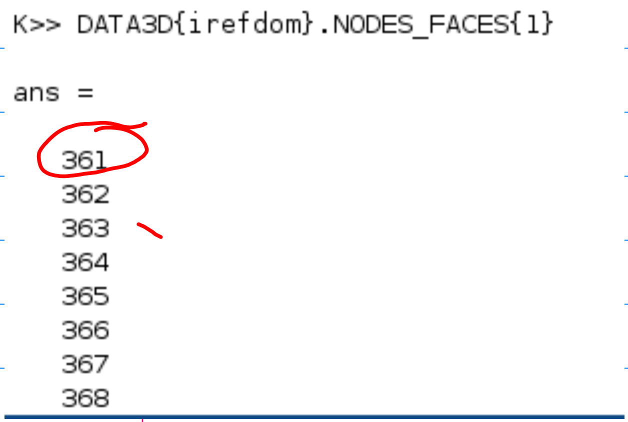
reference domain 1 v



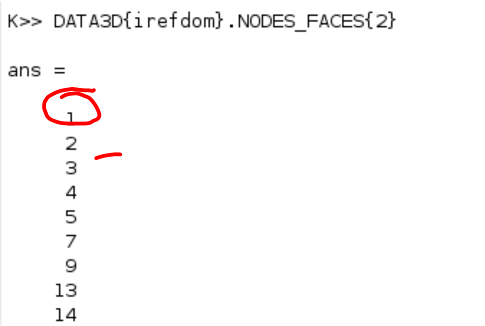
FAce 1



FACE 2



```
K>> DATA3D{irefdom}.NODES_FACES{1}

ans =

   361
   362
   363
   364
   365
   366
   367
   368
```

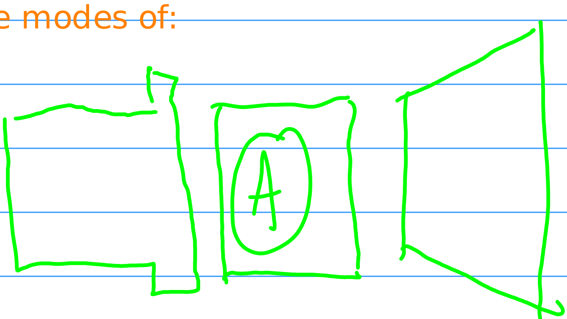


```
K>> DATA3D{irefdom}.NODES_FACES{2}

ans =

     1
     2
     3
     4
     5
     7
     9
    13
    14
```

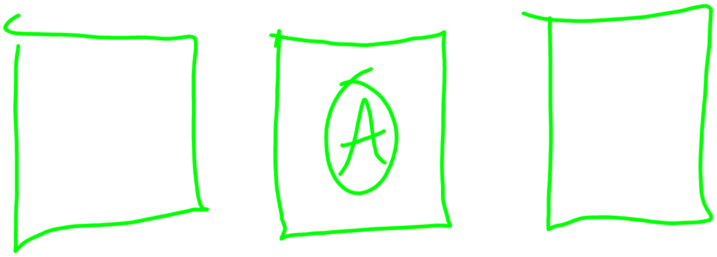
Notice they are paired.
However, if we have several
reference domains, this is not
observed by default... Change it !

OBSERVATION

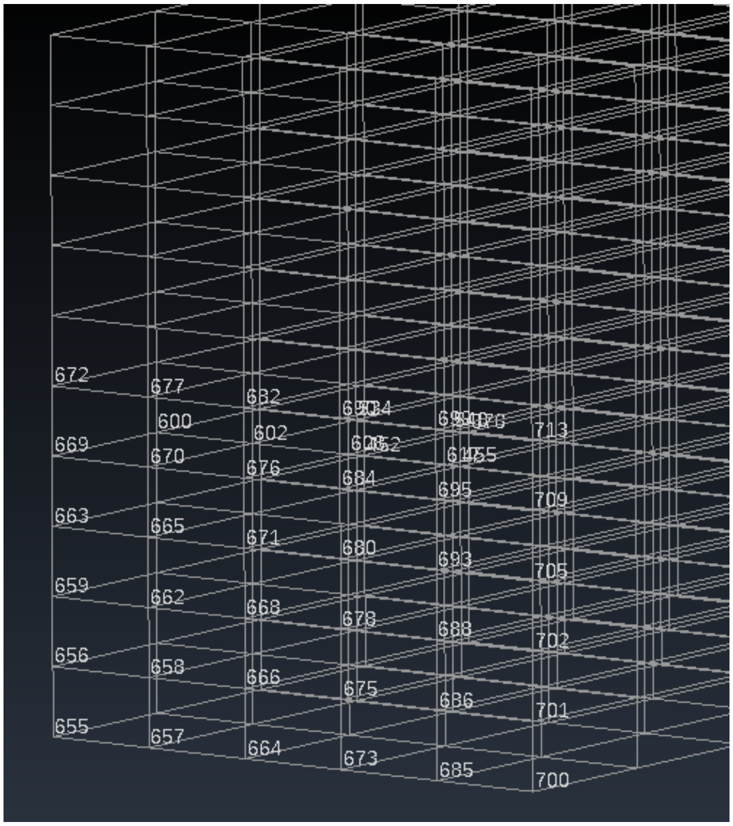
question: Does this approach makes any sense ? From a practical point of view, one has to consider that the parametric training space for a slice is formed, not only by the external actions transmitted by contiguos slices, but by the shape and form of the slices themselves. Thus, the modes of:

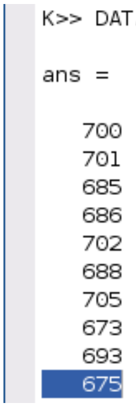


will be necessarily different to the modes of , say,



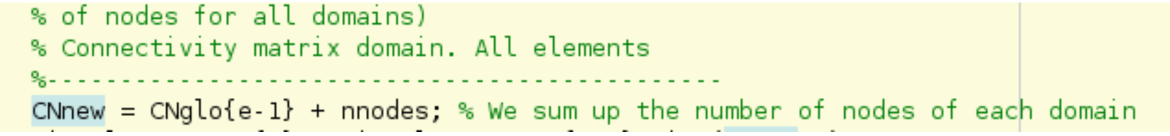
How to handle, in a systematic way, this admittedly thorny issue ?





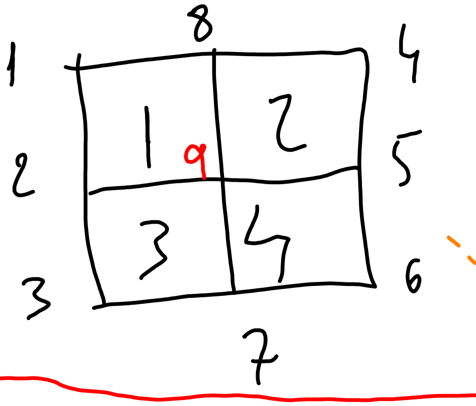
```
K>> DATA3D{irefdom}.NODES_FACES{1}

ans =

   700
   701
   685
   686
   702
   688
   705
   673
   693
   675
```

FACE 1, SLICE REFERENCE 2, AFTER SORTING



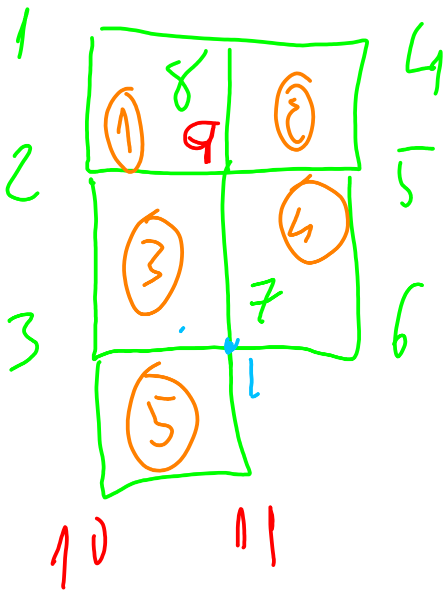
```
% of nodes for all domains)
% Connectivity matrix domain. All elements
%---------------------------------------------
CNnew = CNglo{e-1} + nnodes; % We sum up the number of nodes of each domain
```

In the old code, we used to have the above... And what about now ?

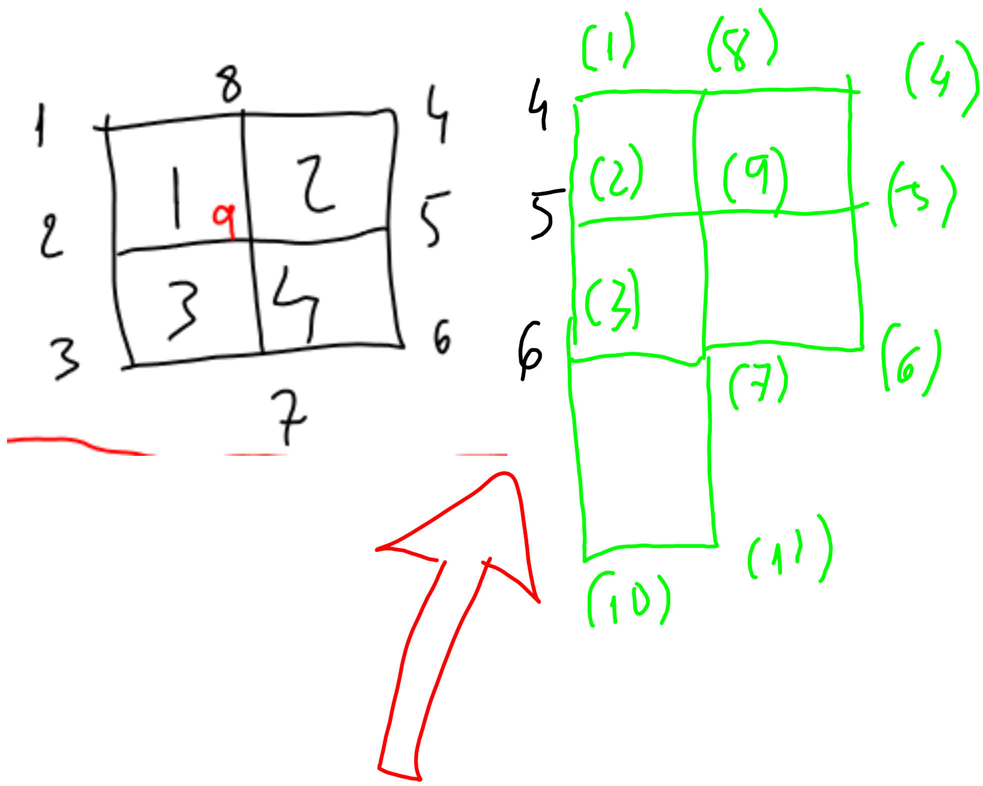


Ref. dom. = 1

$$CN = \begin{bmatrix} 1 & 2 & 9 & 8 \\ 9 & 5 & 4 & 8 \\ 2 & 3 & 7 & 9 \\ 7 & 6 & 5 & 2 \end{bmatrix}$$

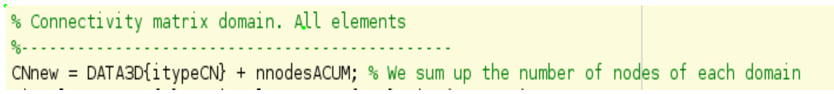


Ref domain = 2



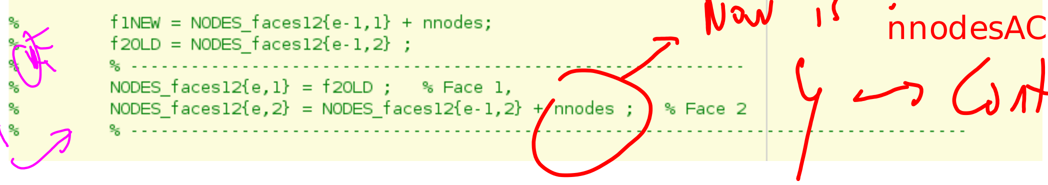
$$CN = \begin{bmatrix} 1 & 2 & 9 & 8 \\ 9 & 5 & 4 & 8 \\ 2 & 3 & 7 & 9 \\ 9 & 7 & 6 & 5 \\ 3, & 10, & 11, & 7 \end{bmatrix}$$

HOW TO DETERMINE THE NEW NUMBERING SCHEME FOR THE INCOMING DOMAIN IS QUITE A THORNY ISSUE....how TO DO IT ? Let us first ignore the merging and just assume that the nodes are different. Therefore:



```
% Connectivity matrix domain. All elements
%-----------------------------------------------
CNnew = DATA3D{itypeCN} + nnodesACUM; % We sum up the number of nodes of each domain
```

VALID for equal slices

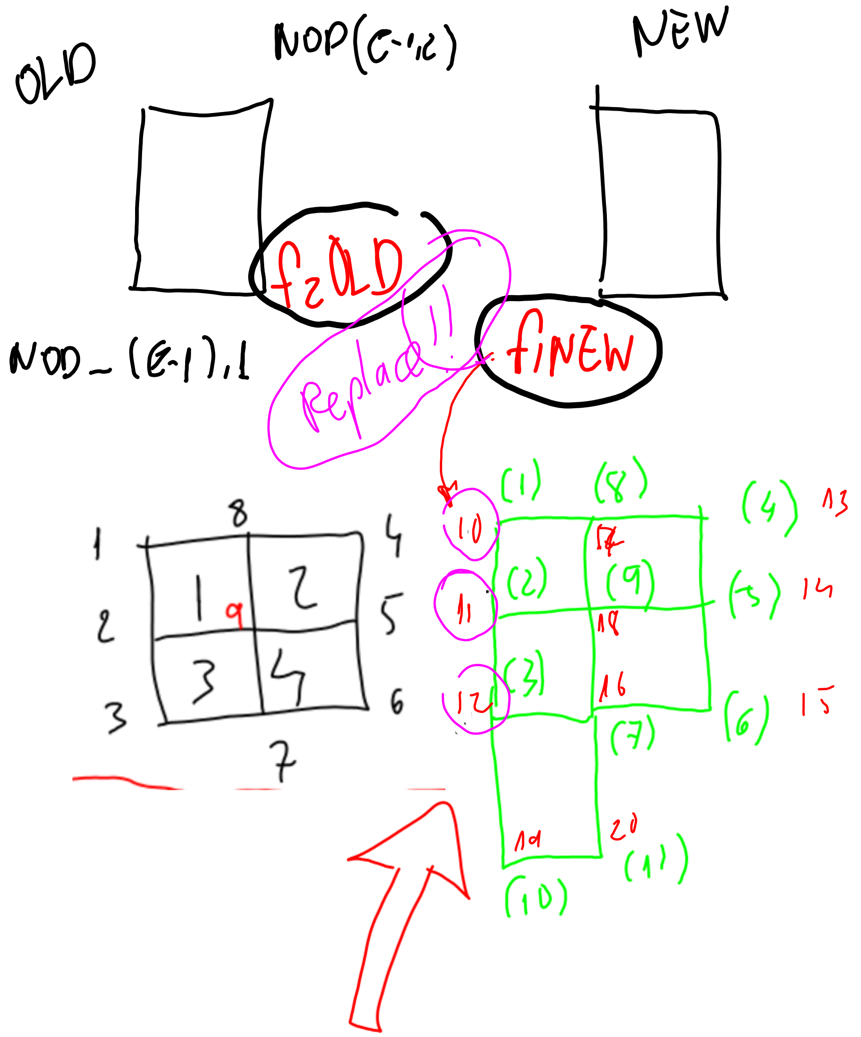


```
%        f1NEW = NODES_faces12{e-1,1} + nnodes;
%        f2OLD = NODES_faces12{e-1,2} ;
%        % ---------------------------------------------------------
%        NODES_faces12{e,1} = f2OLD ;   % Face 1,
%        NODES_faces12{e,2} = NODES_faces12{e-1,2} + nnodes ;   % Face 2
%        % ----------------------------------------------------------------------------------------
```

Now is nnodesACUM

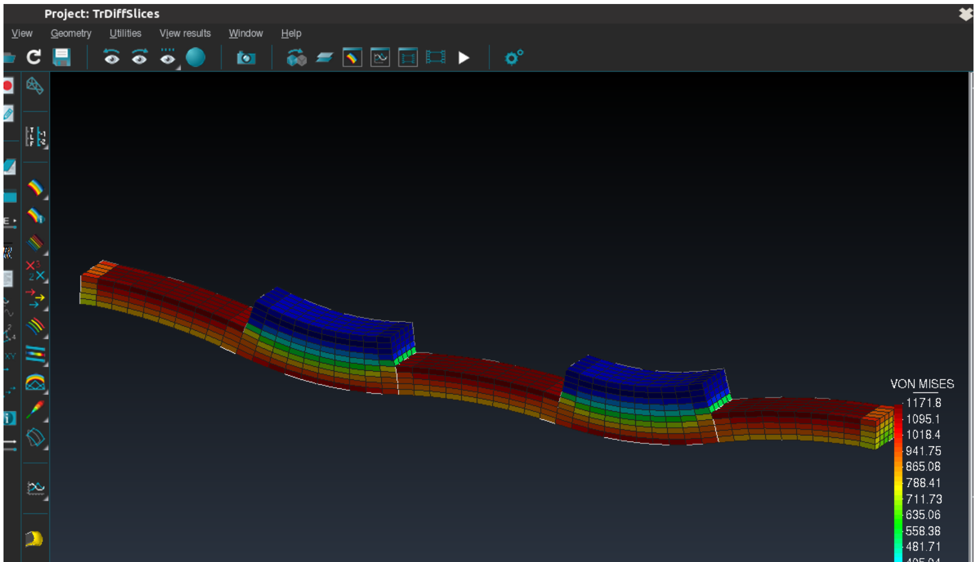
y ⟷ Continuity

What about this excerpt all about ?



HOW TO DETERMINE THE NEW NUMBERING

DONE


Project: TrDiffSlices
View
Geometry
Utilities
View results
Window
Help
VON MISES
1171.8
1095.1
1018.4
941.75
865.08
788.41
711.73
635.06
558.38
481.71