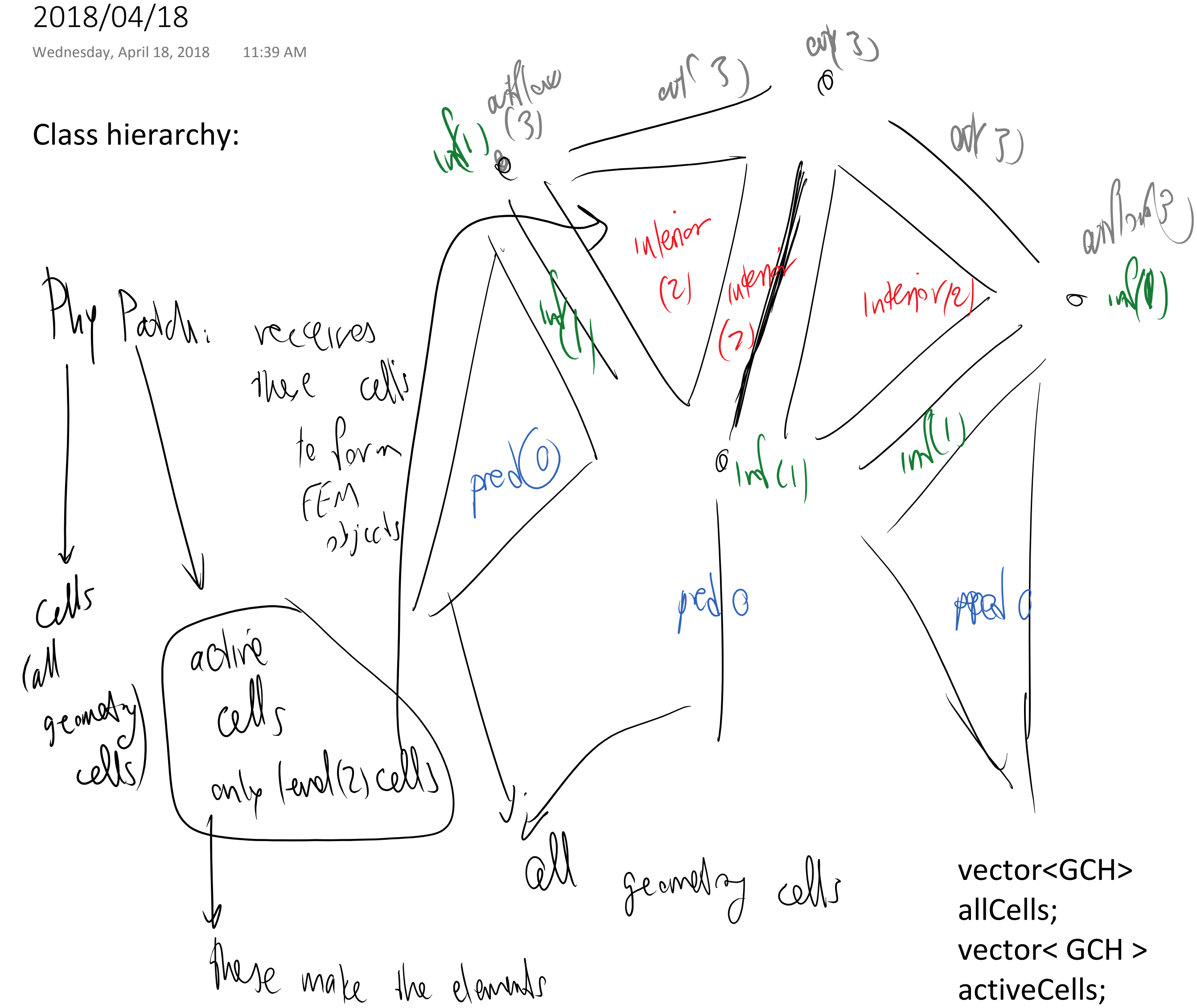
2018/04/18

Wednesday, April 18, 2018 11:39 AM

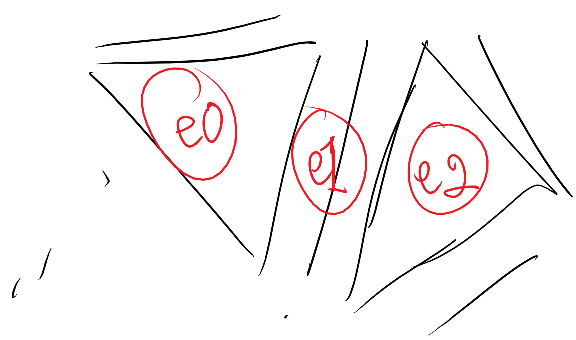
Class hierarchy:



vector<GCH> allCells;
vector< GCH > activeCells;

0. Phy Patch

class PhyPatch: public PhyPatchData

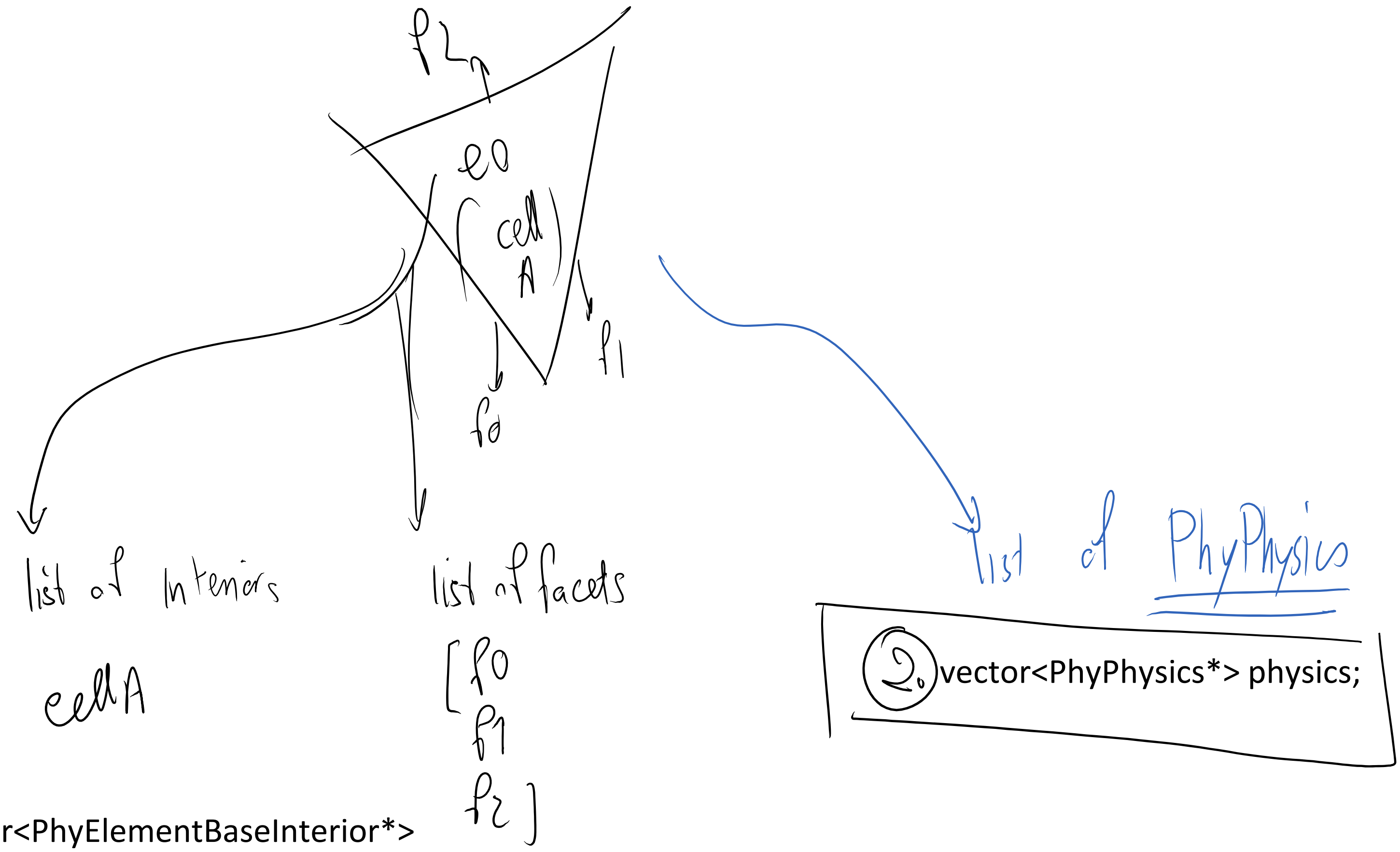


1. { Phy Element> vector

vector<PhyElementBase*> phyElementsBase;

here 3 elements

2. PhyPhysics (each element has a list of physics)   class PhyElementBase



vector<PhyPhysics*> physics;

vector<PhyElementBaseInterior*>
interiorCellsBase

vector<PhyElementBaseFacet*>
facetCellsBase

Members of PhyPhysics and creation of PhyPhysics

For specific physics we need to derive them from a base PhyPhysics class.

There are many specific physics implementations. We use the notation of factory to create them.

PhyPhysics are created by a factory:

Physics/PhysicsFactory.h

```
PhyPhysics* createPhysics(subConfigRef subConRef);
PhyPhysics* createPhysics(subConfigRef subConRef)
{
    PhyPhysics* pp;
    int subConfigIndex = subConRef.subConfigIndex;
    int option;
    switch(subConRef.formulationT)
    {
        case CL:
            option = phyConf->subConf[subConfigIndex]->
            physics_options(0);
            pp = createCLInstance(option);
            break;
        case SL:
            pp = new SLPhysics();
            break;
```

```
// the use of the factory in PhyElement
void PhyElementBase::setPhysics()
{
    num_physics = descProp.subConfigRefs.size();
    physics.resize(num_physics);
    for(int i = 0; i < num_physics; i++)
    {
        physics[i] =
        createPhysics(descProp.subConfigRefs[i]);
        physics[i]->phyLocInElement = i;
        physics[i]->peParent = this;
//          physics[i]->patch = patch;
    }
}
```

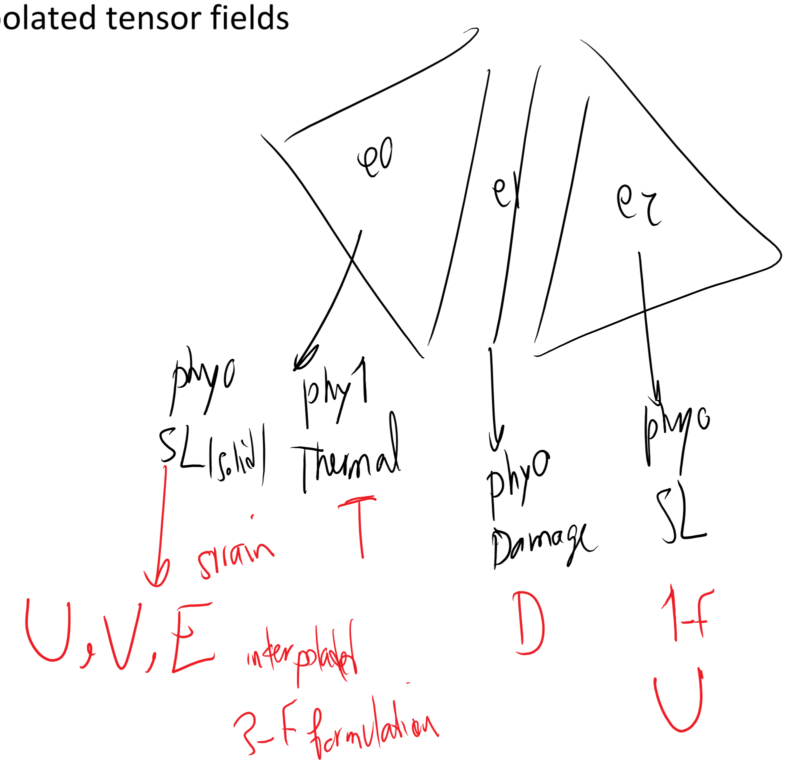
By using this function we create the vector of PhyPhysics inside the element.

What is inside PhyPhysics:

-----

2. PhyPhysics has a vector of interpolated tensor fields

vector<PhyTensorField> pTFields;



4. class PhyTensorField

vTensor<phyField>   physicsFs;

for example U has

U_0, U_1 for 2D elastodynamics

E has E_{00}, E_{11}, E_{01} for 2D ED

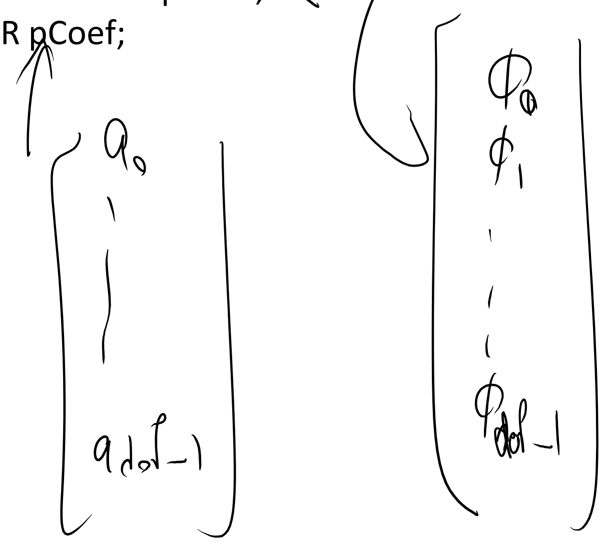
5. phy field Class:

$U_e = \sum_{i}^{\text{dof basis}} \phi_i(x)\, a_i$

class phyField

$$U_0 = \sum_{i=0} \phi_i(x)\, a_i$$

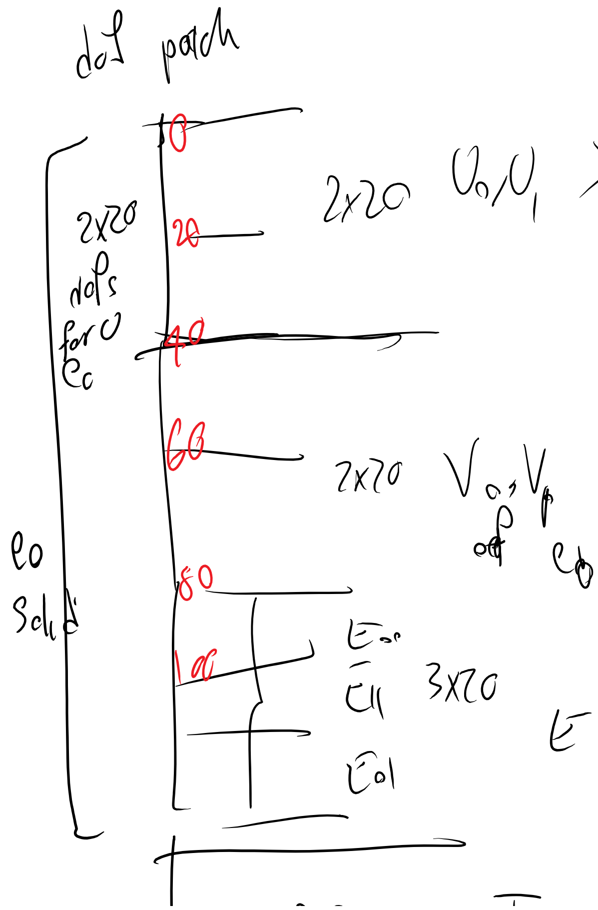
class phyField
PhyBasisElement pBasis;
VECTOR pCoef;

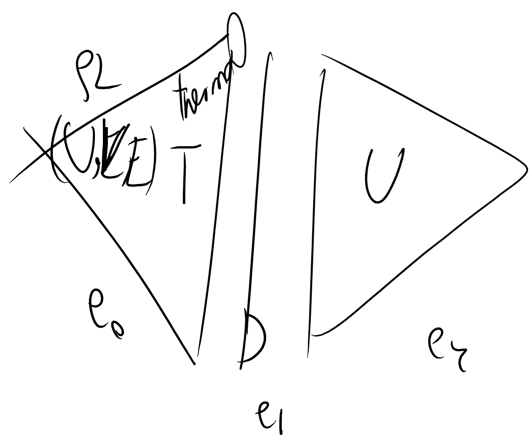


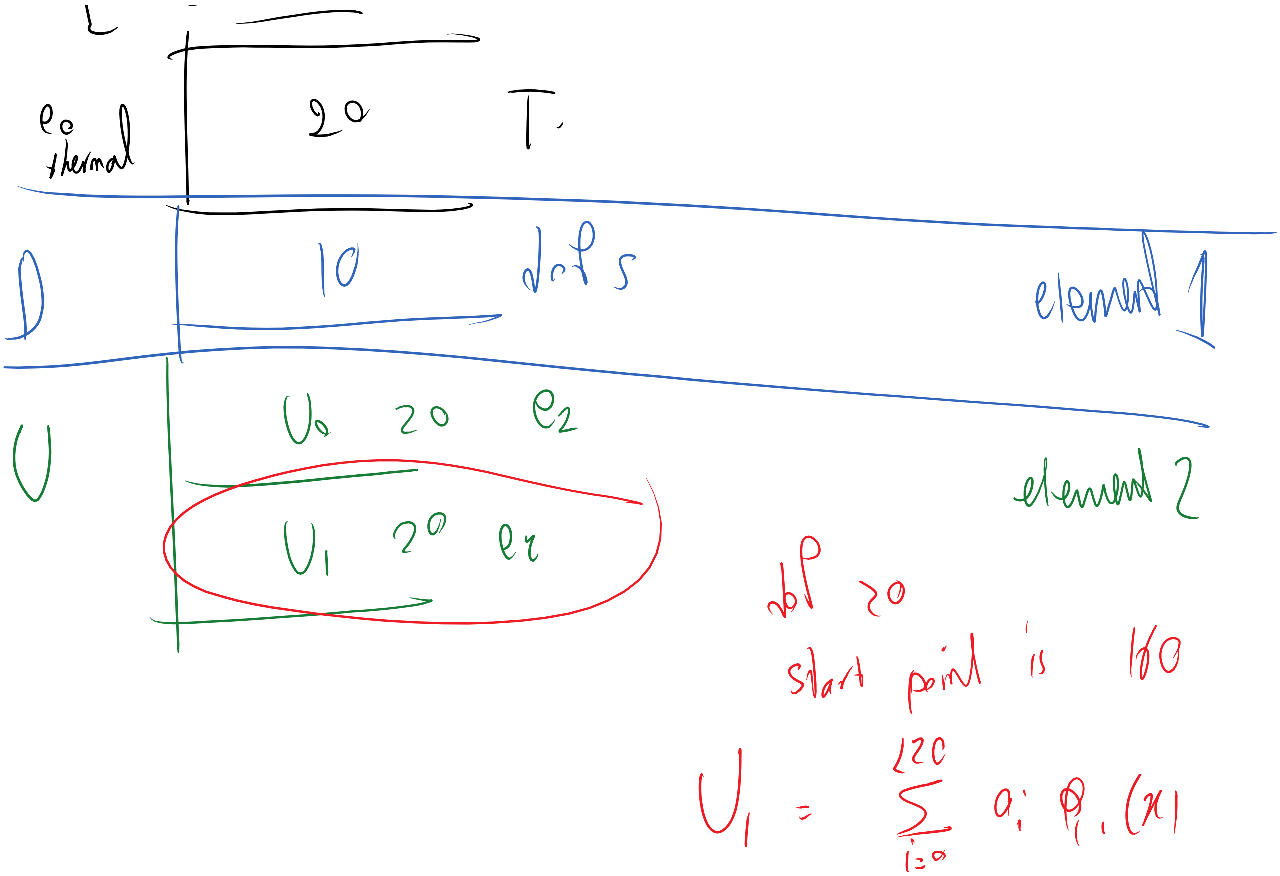
start point is 160

posDof pDof;

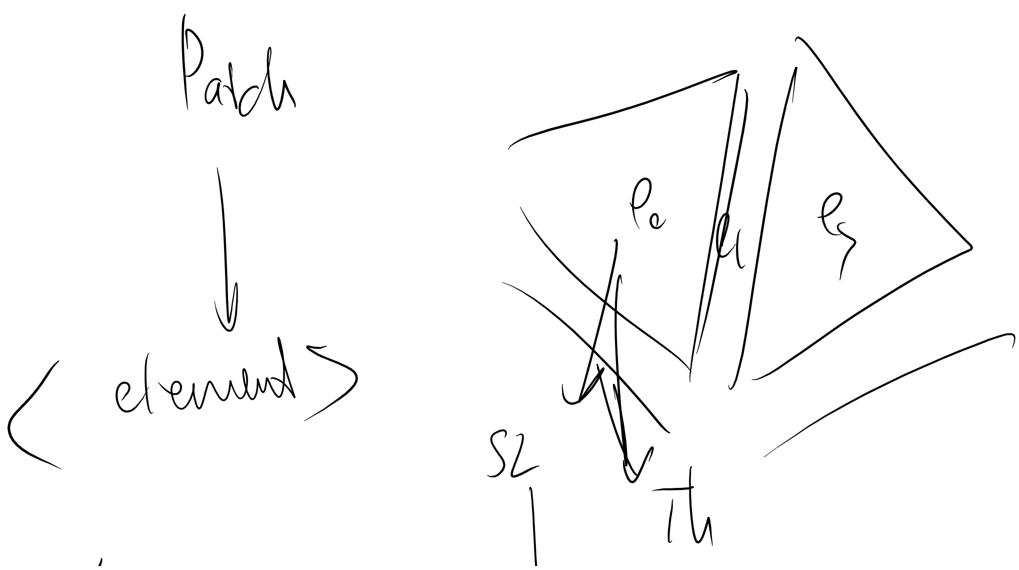
store the location of dofs of different objects in patch Dofs

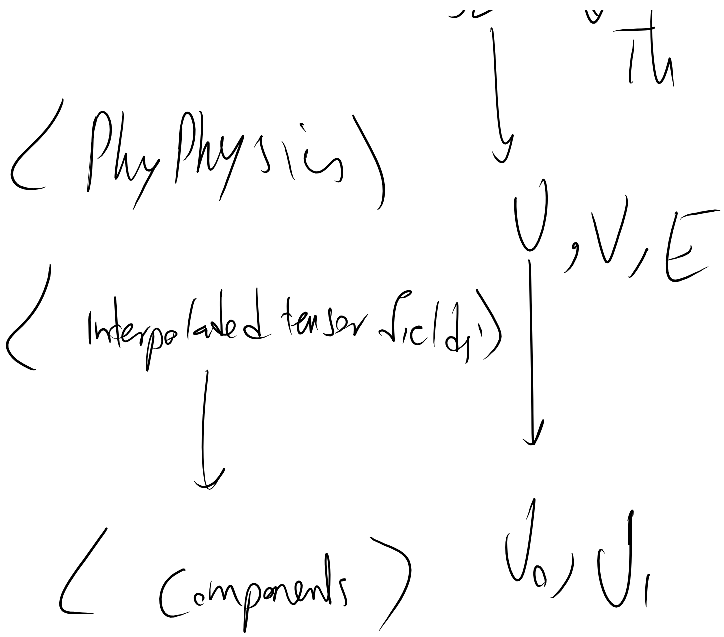




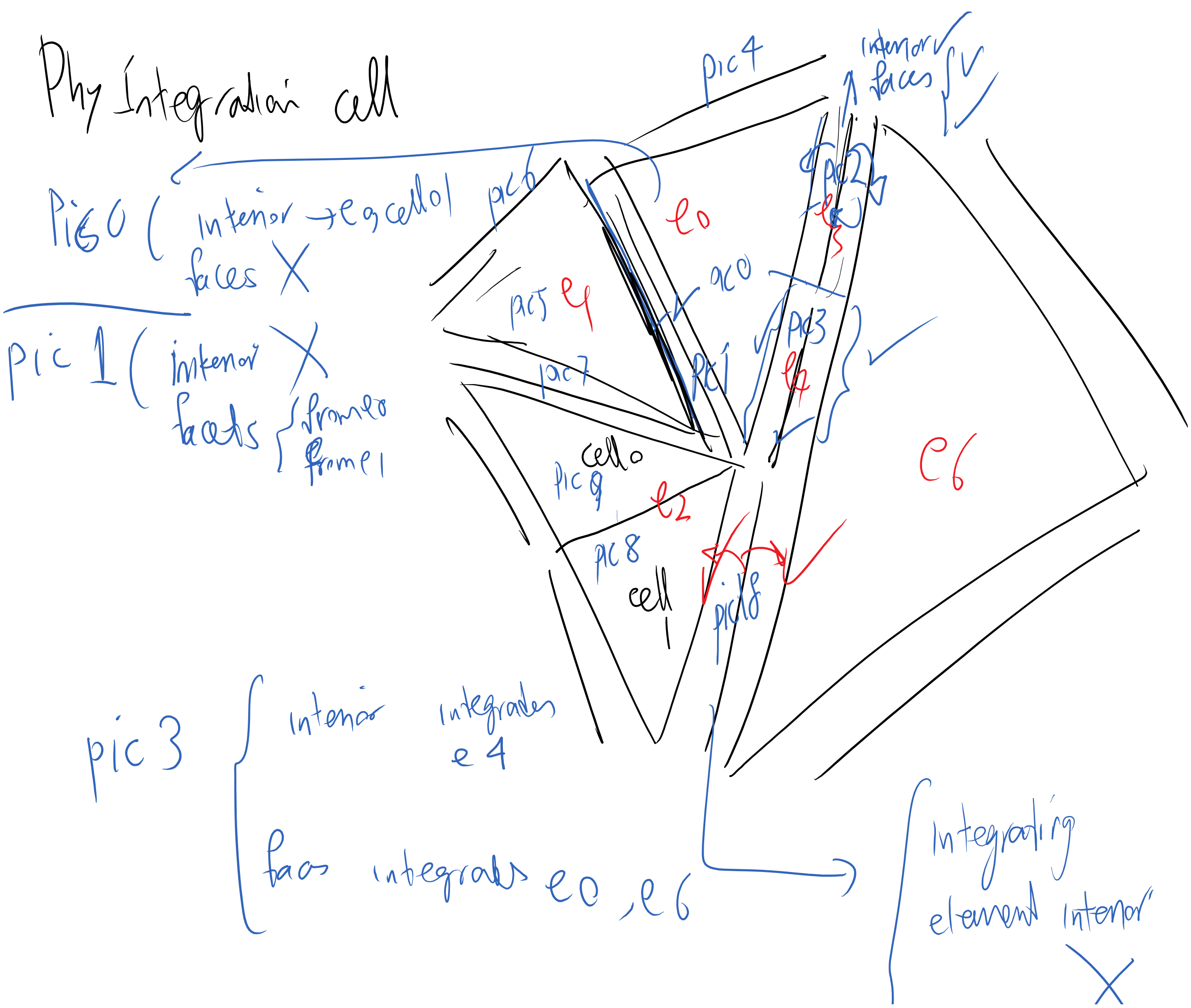


These starting point numbers play the role of dofMap for CFEMs and allow the code to assemble the matrices and vectors to the right place in global stiffness matrix.





Another important hierarchy of the Patch



element interior

integrate any face?
faces of e2, e6

class PhyIntCellBase

PhyInt2EBasePtr interiorIntBase;
bool bInteriorInt;

bb

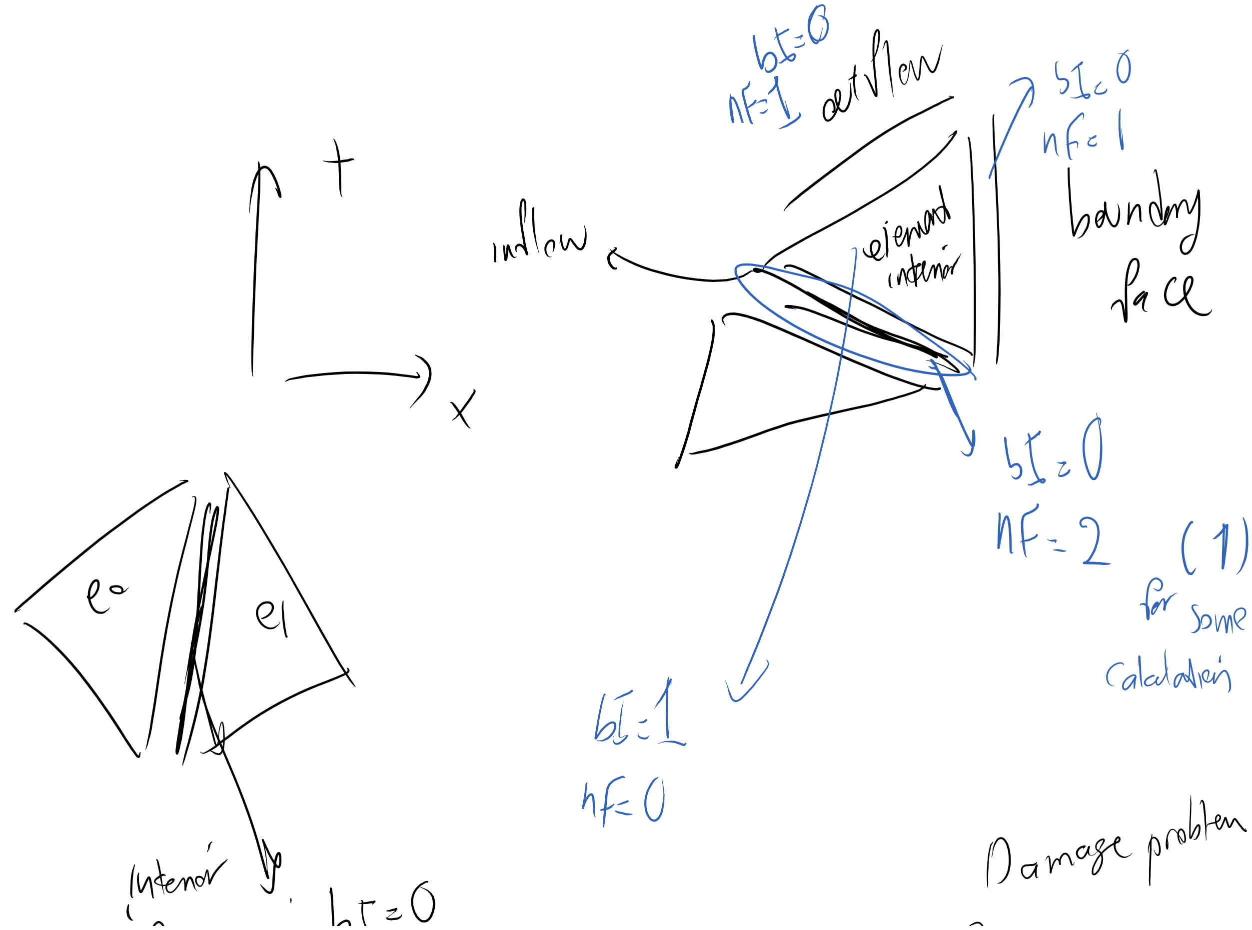
if the cell is integrating any interior

vector<PhyInt2EBasePtr> facetIntBase;

size of this

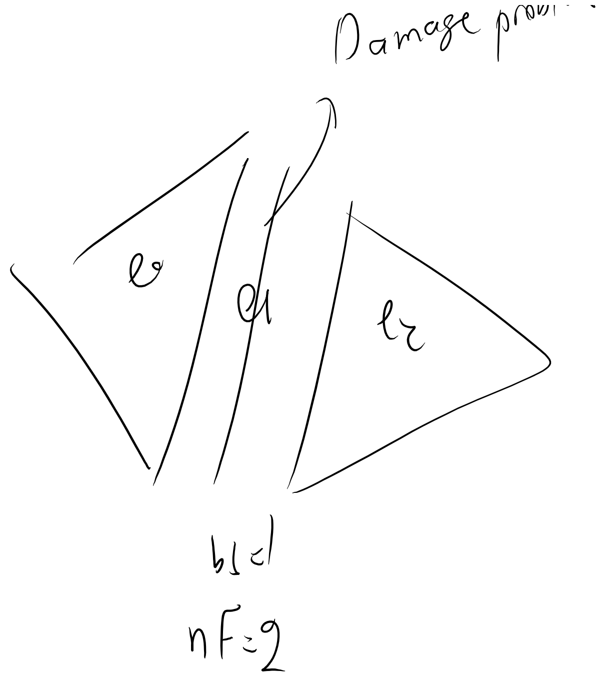
nF

how many faces it integrates



Interior face $bf=0$ $nF=2$

Most general



PhyElements and PICs are stored in PhyPatch

```
Class PhyPatch
....
vector<PhyElementBase*> phyElementsBase;
int num_elementsBase;
```
} elements

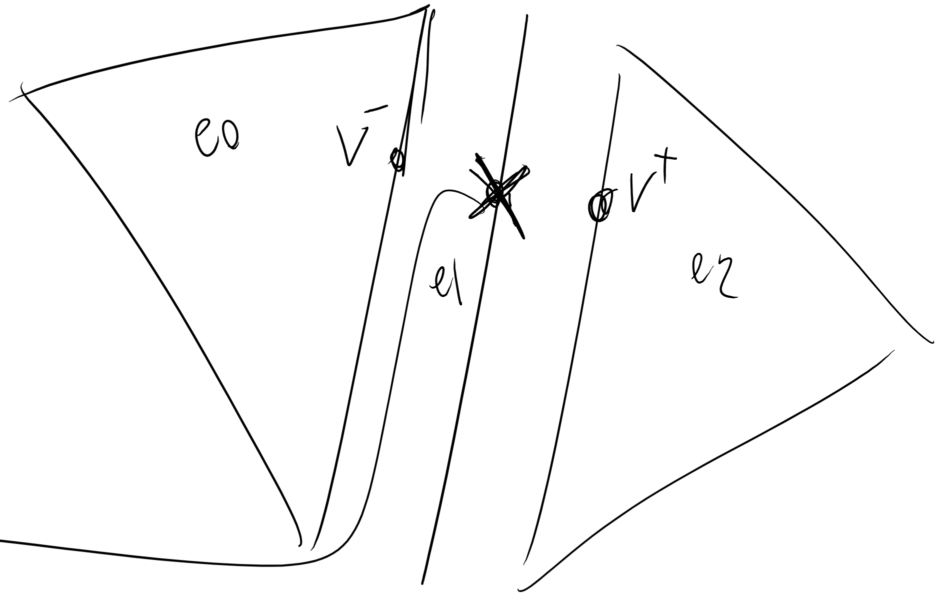
```
vector<PhyIntCellBase*> phyIntsBase;
int num_phyIntsBase;
```
} integration cells

Storage members:

Have to have a way to name tensors.

```
class PhyFldC
...
phyFld phyF;
compT  cT;
```
phyF → U, V, S, E, ...

cT → Val, DT, Dx ...

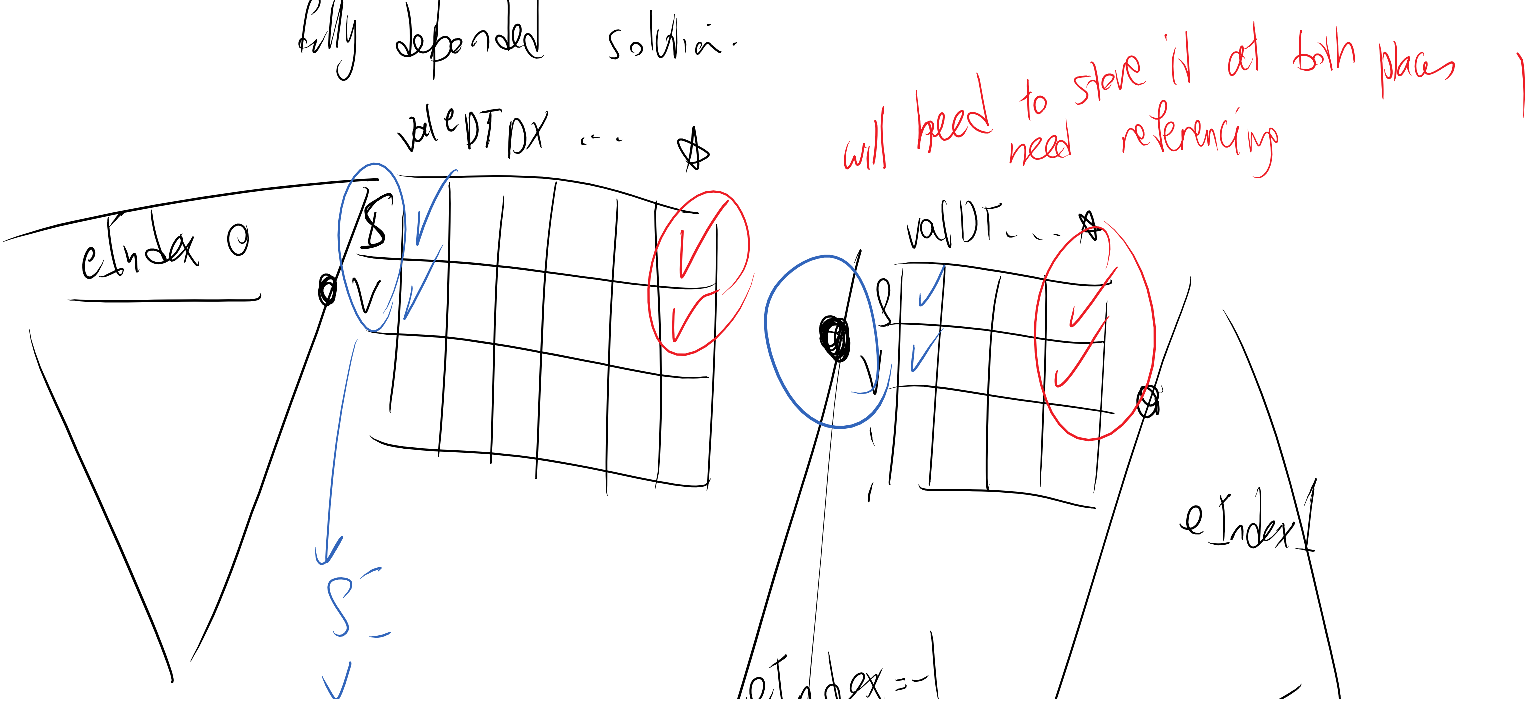


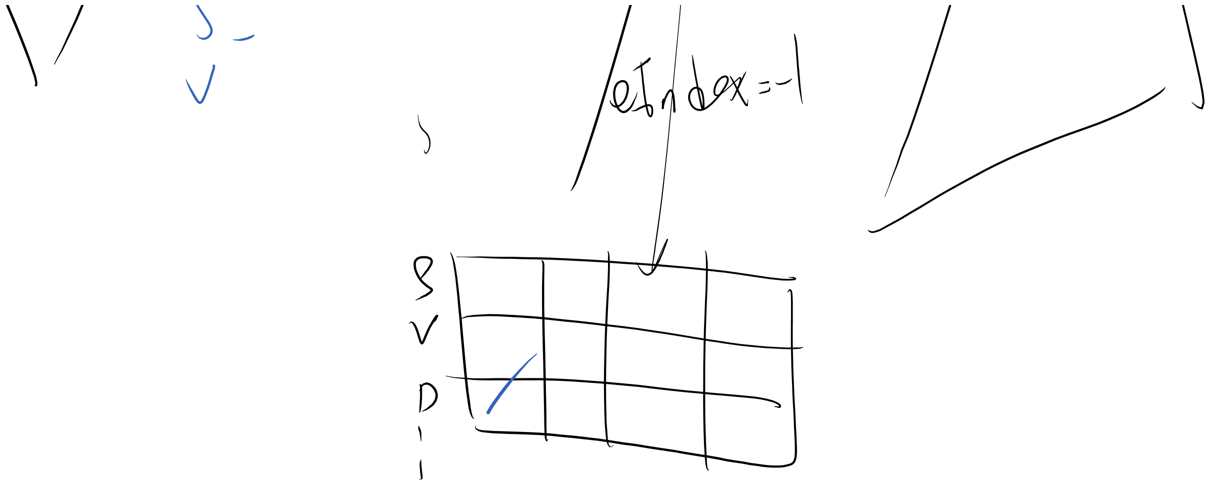
$t_i = S_{ij} n_j$

$$t_0^{\star} = \left(\frac{t_0^- Z^+ + t_0^+ Z^-}{Z^- + Z^+} + \frac{Z^- Z^+}{Z^- + Z^+} \left(V^+ + V^- \right) \right) (1 - D)$$

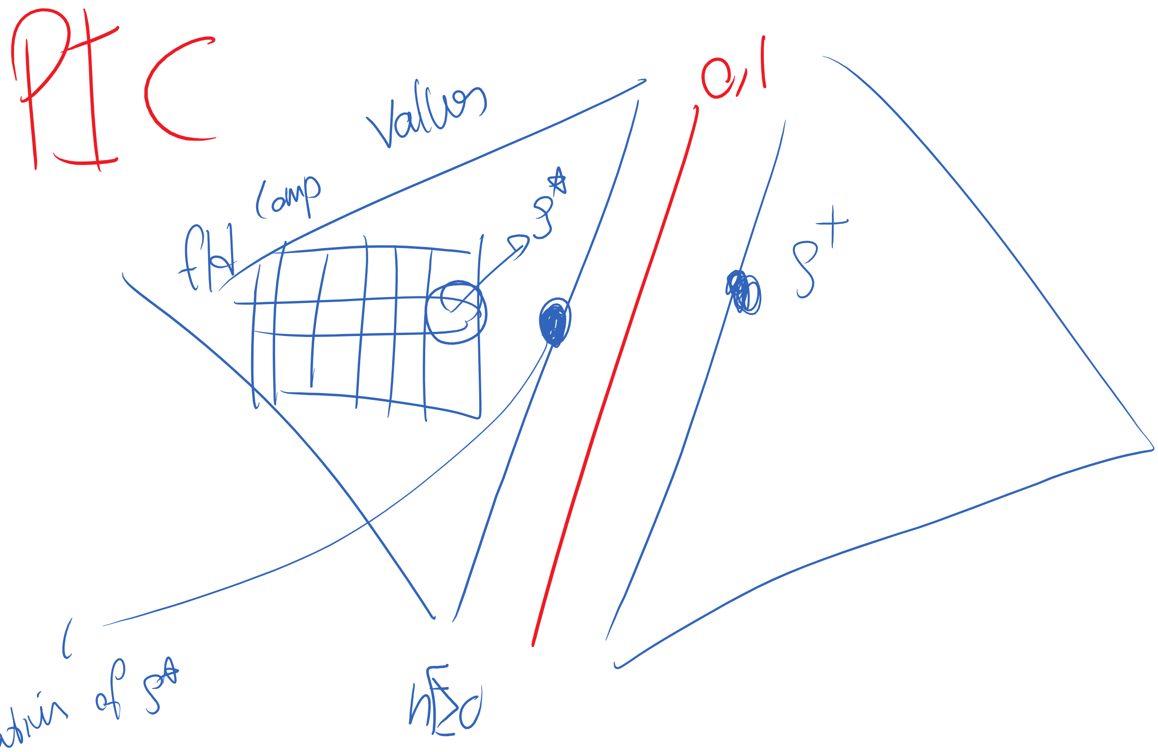
direction

we go from a fully bonded ($D = 0$) to fully debonded solution.





PIC

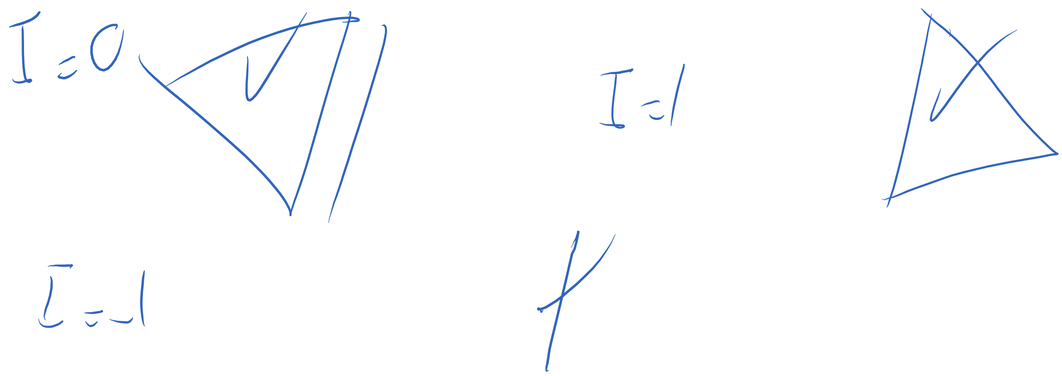


derivation of ρ^* are w.r.t what?

$$\frac{\partial \rho^*_\theta}{\partial a_I} = \frac{Z^-}{Z^- + Z^+}(1-1)\frac{\partial \rho_0^+}{\partial a_I} - \frac{Z^-}{Z^+ + Z^+}\frac{\partial D}{\partial a_I}\frac{\rho_0^+}{\partial a_I} \cdots$$

$$+ \frac{Z^+}{+}(1-1)\frac{\partial \rho_0^-}{} - \frac{Z^-}{}\frac{\partial D}{}\frac{\rho_0^-}{}$$

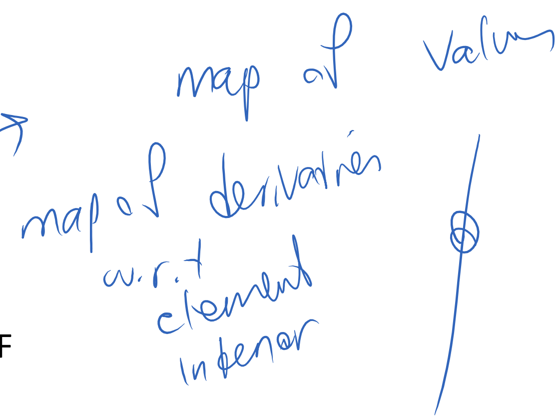
$$+ \frac{t}{z^t + z}(1+D)\frac{\partial J_o}{\partial a_I} - \frac{z}{z + z^t}\frac{\partial D}{\partial a_I}\frac{J_o}{\partial a_I}$$



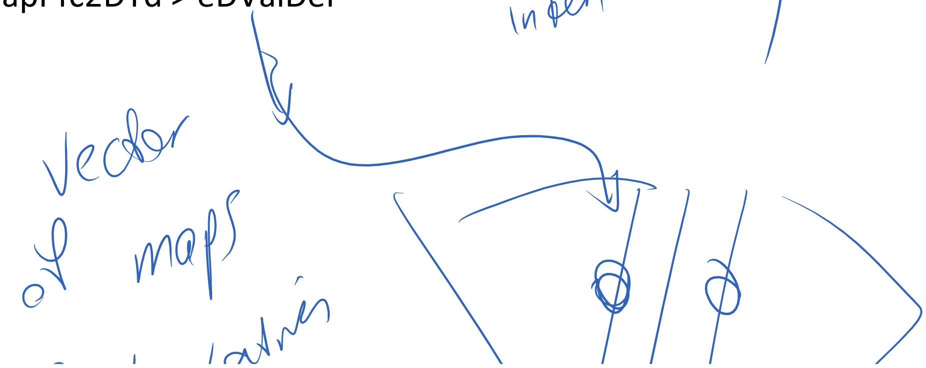
So the challenge is that any component of a tensor can depend on all elements that are present at a PIC

The storage for all values and shapes for one quadrature points for one of the elements:

class PhyFieldElement



mapPfc2Td eVals
mapPfc2DTd eDValDeI
vector< mapPfc2DTd > eDValDeF



of derivatives
w.r.t elements
attached by facets