

Operator ASSE_MAILLAGE

1 Goal

To assemble two grids to form new.

The two grids to be assembled can come from the operators: LIRE_MAILLAGE [U4.21.01], DEFI_MAILLAGE [U4.23.01], ASSE_MAILLAGE [U4.23.03],...

I.e. they can contain ordinary meshes or super-meshes carrying macro - elements.

Product a structure of data of the type `grid`.

2 Syntax

```
mac (sd_maillage) = ASSE_MAILLAGE (  
  
    # names of two grids to be assembled:  
    ♦ MAILLAGE_1 = ma1, [grid]  
    ♦ MAILLAGE_2 = ma2, [grid]  
  
    # type of operation to be realized on the two grids  
    ♦ OPERATION = / 'JOINING', #to restick  
                  / 'SUPERIMPOSES', #to superimpose  
                  / 'SOUS_STR', #pour to assemble  
                  #grids containing  
                  #macronutrients  
  
    # If OPERATION=' COLLAGE'  
    ♦ COLLAGE=_F (  
        ♦ GROUP_MA_1 = gma1, [gr_maille]  
        ♦ GROUP_MA_2 = gma2, [gr_maille]),  
    )
```

3 Operands MAILLAGE_1 and MAILLAGE_2

- ◆ MAILLAGE_1 = ma1, MAILLAGE_2 = ma2,
ma1 and ma2 are the names of the two grids to be assembled.

4 Operand OPERATION = 'SUPERIMPOSES'

With choice 'SUPERIMPOSES', all the entities (meshes, nodes, groups of meshes and groups of nodes) of two grids are preserved. The only problem to be regulated relates to the name as of these entities (possible conflict of names because them two grids can contain, for example, of the same nodes name).

To solve these possible problems of names:

- 1) The nodes and the meshes are always famous. That wants to say that the user cannot know the name of the nodes and meshes of the grid result. It will have in general to use (what is always advised) the names of groups of meshes and groups of nodes. Or it will have to print the grid to know the selected names.
- 2) The names of the groups of nodes and the groups of meshes are preserved within the limit of the possible one. If two groups of meshes (or two groups of nodes) have the same name in two grids, the group coming from the 2nd grid is famous automatically and the name change is indicated in the file of message.

5 Operand OPERATION = 'JOINING'

The operation 'JOINING' is used to connect two grids which would have been with a grid independently in two distinct files (for example by two different teams).

If them two grids are not coherent on their interface (different discretization), the user will not have another choice only to use the functionality `AFFE_CHAR_*/LIAISON_MAIL` to bind the degrees of freedom of two grids which will remain topologically disjointed. It will have to then use the operation 'SUPERIMPOSES'.

If on the other hand, them two grids were designed to be restuck, it will use the operation 'JOINING'. For that, it will have to take the precaution to name them two groups of meshes (of interface) which will make it possible to restick them two grids. Moreover these two groups of meshes must be geometrically coincidents. The user will write then:

```
OPERATION=' COLLAGE', COLLAGE= _F (GROUP_MA_1=' gma1', GROUP_MA_2=' gma2',),
```

Groups of meshes `gma1` and `gma2` will then be amalgamated. More precisely:

- 1) meshes of `gma1` and `gma2` will be removed,
- 2) nodes of `gma2` will remain (but will be orphan),
- 3) nodes of `gma1` will connect them two grids.

Before amalgamating them two groups of meshes `gma1` and `gma2`, it code will check that the nodes of these two groups of meshes are well coincidents (with a tolerance of 0,001 time the length of smallest edge two grids).

Conflicts of names of the entities of two grids are regulated in the same way that for the operation 'SUPERIMPOSES' (see above).

6 Operand OPERATION = 'SOUS_STR'

To mix in the same model of the ordinary finite elements and macronutrients (or substructures), it is necessary to have a grid containing at the same time ordinary meshes and super-meshes. The operator ASSE_MAILLAGE / OPERATION = 'SOUS_STR' allows to constitute this "mixed" grid by assembling an ordinary grid (or mixed) and a grid containing of the super-meshes (coming from DEFI_MAILLAGE).

The direction of the assembly is the following:

- **All** entities of two grids arguments `ma1` and `ma2` (meshes, super-meshes, nodes, group of meshes and group of nodes) are recopied in the grid result: `mac`.
- The only shared entities are them **nodes of same names**. These are the nodes which make it possible to assemble them two grids.
- Treatment of the entities bearing the same name:
 - nodes: the nodes of the second grid bearing an existing name in the first grid are not added: it is supposed that they are the same ones. The coordinates of the preserved node are those of the node of the first grid. An alarm is emitted when the distance between the two confused nodes is higher than:

$$10^{-6} \cdot d_refe$$

where d_refe is a length characteristic of the grid:

$$d_refe = \max(d(O, N))$$

where $d(O, N)$ is the distance from the node N at the origin of the total reference mark.

- meshes (or super-meshes): if grid `ma2` of the same meshes name contains than meshes of the first grid `ma1`, it code stop in fatal error.
- group of meshes (or groups of nodes): if grid `ma2` a group contains of of the same meshes name than a group of mesh of `ma1`, this one is ignored and it code a message of alarm emits.