Operator DEFI_MAILLAGE

1 Drank

To define a mesh using macro-elements.

This command makes it possible to define a new mesh from static or dynamic macro-elements. This new mesh (containing only the geometrical supports of the macro-elements) can then be “assembled” with another mesh (container for example of meshes “classical” thanks to the command ASSE_MAILLAGE [U4.23.03] and the option specific to the substructuring.

Product a data structure of mesh type .

2 Syntax

```
my (mesh) = DEFI_MAILLAGE (
    ♦DEFI_SUPER_MAILLE   =  (_F    (
       ♦MACR_ELEM=l_se                ,                      [l_macr_elem_*]
       ◊SUPER_MAILLE =l_mail          ,                 [l_maille]
       ◊   |  ◊TRAN=/                  (tx, ty), or (tx, ty, tz),  [l_R]
                                /   (0. , 0.) or (0. , 0. , 0.),   [DEFAULT]
          |  ◊ANGL_NAUT=/(a)             ,  or (has, B, c),       [l_R]
                                /   (0.), or (0. , 0. , 0.),    [DEFAULT]
             ◊CENTER=/                  (px, py) or (px, py, pz),   [l_R]
                                /   (0. , 0.), or (0. , 0. , 0.),  [DEFAULT]
                          ),),
    ◊   |      RECO_GLOBAL= (_F  (
                               ♦/TOUT=' OUI',
             /SUPER_MAILLE     =l_maille       ,              [l_maille]
          ◊   |  CRITERE=/        "ABSOLU"  ,
                               /   "RELATIF" ,              [DEFAULT]
             |  PRECISION=/prec                ,              [R]
                               /1.D-3          ,              [DEFAULT]
                            ), ),
       |  RECO_SUPER_MAILLE =  (_F    (
          ♦SUPER_MAILLE=l_mail                  ,                  [l_maille]
          ♦GROUP_NO=l_gno                       ,                  [l_group_no]
          ◊/OPTION       =   ' GEOMETRIQUE'  ,                     [DEFAULT]
                ◊   |  CRITERE=/         "ABSOLU"  ,
                                        /   "RELATIF" ,          [DEFAULT]
                   |  PRECISION=/prec              ,             [R]
                                        /1.D-3     ,             [DEFAULT]
             /OPTION     =   ' NOEUD_A_NOEUD',
             /OPTION     =   ' INVERSE',
                            ),),
    ◊DEFI_NOEUD    =_F        (
          /♦TOUT         =          ' OUI'    ,
             ◊PREFIXE   =pref             ,                    [kN]
             ♦INDEX     =             (DM, Fm, dn, fn),          [l_I]
          /♦NOEUD_FIN=no_fin                    ,                    [node]
             ♦SUPER_MAILLE=mail                 ,                    [mesh]
             ♦NOEUD_INIT=no_ini                 ,                    [node]
                         ),),
    ◊DEFI_GROUP_NO    = _F   (
                                  /♦/TOUT=' OUI'   ,
                /SUPER_MAILLE =mail          ,           [mesh]
             ◊PREFIXE      =pref           ,                [kN]
             ♦INDEX      =     (DM, Fm, dn, fn),          [l_I]
          /♦GROUP_NO_FIN=gno_fin              ,                [group_no]
             ♦SUPER_MAILLE=mail               ,                [mesh]
             ♦GROUP_NO_INIT=gno_ini     ,                      [group_no]
                         ),),
                                    )
```

3 General information

In the documentation of this command, one will speak about:

- macro-element: object of the macr_elem_stat `type` or `macr_elem_dyna`,
- super-mesh: geometrical entity supporting one macro_élément,
- initial **mesh** when one indicates the mesh which was used to generate a macro - element,
- final **mesh** to indicate the mesh produced by this command.

By extension these adjectives **initial/final** will apply to the entities attached to the meshes: node, mesh, nodes group.

Practically, to build the final mesh:

- one defines the superones while positioning in the space (2D or 3D) of the macro - existing elements (the same macro-element can generate several super-meshes),
- one resticks the superones between them,
- one re-elects, if it is wanted it, some nodes,
- one creates, if it is wanted, certain nodes groups.

Note:

One can note that the mesh created by this command is made only of super - meshes. One thus cannot (for example), to draw it with the post - usual processors. Possibilities of curing it will be able to exist with command `DEFI_SQUELETTE` *[U4.24.01].*

To mix of the finite elements "classical" and of substructures, it is necessary to use the operator of "concatenation" of meshes [U4.23.03]: `mag = ASSE_MAILLAGE (MAILLAGE= (m1, m2))`

A mesh resulting from operator `DEFI_MAILLAGE` contains:

- the superones,
- nodes,
- nodes groups.

The superones are defined by translation/rotation of macro-elements.

As a "classical" mesh, a super-mesh is entirely defined by the list of its nodes. The coordinates of the nodes of meshes are those of the external nodes of the macro - elements transformed by the geometrical transformation: translation, rotation…

If one does not carry out a resticking (cf `RECO_GLOBAL` / `RECO_SUPER_MAILLE`), the mesh has as many nodes as the sum of the nodes super-meshes.

Convention C1:

When one "resticks" the superones, one eliminates some nodes. By convention, during an elimination of coinciding nodes, one preserves the node (and thus its coordinates) which comes from the first mesh of the list `l_mail` (cf `RECO_GLOBAL` / `RECO_SUPER_MAILLE`).

As in any Aster mesh , the nodes **are named**. By defaults, the names of the nodes are given by the program in the form: `Nijk` where `ijk` is a number ranging between `1` and `999999.9`.

Key words `DEFI_NOEUD` and `DEFI_GROUP_NO` make it possible to the user some nodes to re-elect and to define nodes groups.

4 Operands

4.1 Key word DEFI_SUPER_MAILLE

♦DEFI_SUPER_MAILLE =

This key word factor makes it possible to define super-meshes mesh using the macro-elements.

4.1.1 Operand MACR_ELEM

♦MACR_ELEM = l_se

l_se is the list of the names of the macro-elements which will generate meshes.

4.1.2 Operand SUPER_MAILLE

◊SUPER_MAILLE = l_mail

l_mail is the list of the names which one wants to give to meshes. This argument is optional. In his absence, one will meshes give to the names macro-elements (this is obviously impossible if one wants to use several times the same macro-element).

4.1.3 Geometrical operands of transformations

◊ | ◊TRAN =

This key word defines the translation to be applied to the macro-element:

- if one is in **2D**, one expects 2 realities: (tx, ty),
- if one is in **3D**, one expects 3 realities: (tx, ty, tz).

| ◊ANGL_NAUT =

◊CENTER =

These key words define rotation to be applied to the macro-element.

If one is in **2D**, one expects 3 realities:

- has is the angle (in degrees) of rotation in the plane for ANGL_NAUT,
- px and py is the coordinates of the center of rotation for CENTER.

If one is in **3D**, one expects 6 realities:

- has, B, C are the nautical angles (α, β, γ) of rotation (in degrees). (Cf operator AFFE_CARA_ELEM [U4.42.01]) for ANGL_NAUT,
- px, py and pz are the coordinates of the center of rotation for CENTER.

Notice important:

*It is known that the order of the keywords is not significant for Aster. The operation of translation/rotation is conventionally made in the order rotation **then** translation. These two operations do not commutate in general.*

4.2 Key word RECO_GLOBAL

```
◊  |  RECO_GLOBAL =
   ♦/TOUT              = "OUI',
     /SUPER_MAILLE  = l_maille,
   ◊  |  CRITERE=/        "ABSOLU"  ,
                       /   "RELATIF" ,              [DEFAULT]
     |  PRECISION=/prec          ,
                       /1.D-3   ,                   [DEFAULT]
```

This key word make it possible automatically **to restick** a set of super-meshes (indicated by the key word SUPER_MAILLE or the key word TOUT) with a geometrical criterion of proximity: 2 nodes of 2 super-meshes different m1 and m2 will be confused if the distance which separates them is:

```
<prec                          (CRITERE = "ABSOLU"),
<prec*min  (D (m1), D (m2))    (CRITERE = "RELATIF").
```

where D (semi) notes the smallest distance between 2 nodes of the semi super-mesh.

Note:

Two nodes of the same mesh will never be restuck.

If a mesh contains one node, it is necessary to use CRITERE = "ABSOLU".

4.3 Key word RECO_SUPER_MAILLE

◊RECO_SUPER_MAILLE =

This key word factor makes it possible to restick "**with the super-meshes** hand" certain indicated by the user. The superones that one can restick are those which were defined by the DEFI_SUPER_MAILLE.On key word then resticks the superones via nodes groups. To say what one wants to restick it is thus necessary to give of the couples (mesh, nodes group (of the initial mesh)).

Note:

When one gives a couple (mesh , nodes group), one indicates the list of the nodes of the nodes group which are external for the macro-element which defines the super-mesh. It is in fact the intersection of the nodes group and edge of under - structure. This list is ordered like the initial nodes group.

In theory, when one resticks 2 via 2 nodes groups meshes, all the indicated nodes must be restuck (cf the convention chosen by key word OPTION). An alarm message will be transmitted if it is not the case.

4.3.1 Operands SUPER_MAILLE / GROUP_NO

◊SUPER_MAILLE =

One gives here the list of meshes to restick. In general, one resticks meshes the 2 by 2.

For the "corners", it can be pleasant to restick all the meshes convergent ones in only once (for example the 4 super-cubic ones which divide the same edge).

◊GROUP_NO =

One gives the list of the nodes groups here to be restuck. This list is of the same length than the list of meshes.

4.3.2 Operand OPTION

◊OPTION =

This key makes it possible to choose the convention of resticking of the lists of nodes defined by the nodes groups.

- "GEOMETRIQUE" :

  The program will confuse the nodes by considerations of geometrical proximity. (Cf key word: RECO_GLOBAL)

- "NOEUD_A_NOEUD"/"INVERSE":

  Soit : $G1 = \{ A1, B1, C1 \}$
  $G2 = \{ A2, B2, C2 \}$
  $G3 = \{ A3, B3, C3 \}$

  If OPTION = "NOEUD_A_NOEUD" , GROUP_NO = (G1, G2, G3)

  on va recoller : A1 avec A2 avec A3
  B1 avec B2 avec B3
  C1 avec C2 avec C3

  If OPTION = "INVERSE" , GROUP_NO = (G1, G2, G3)

  on va recoller : C1 avec A2 avec A3
  B1 avec B2 avec B3
  A1 avec C2 avec C3

  Attention:

  For option "INVERSE" , only the first nodes group of the list of the GROUP_NO "is turned over".

4.4 Key word DEFI_NOEUD

◊DEFI_NOEUD =

This key word factor makes it possible to re-elect whole or part of the nodes of the mesh.

4.4.1 Operands TOUT / PREFIXE / INDEX

| ♦TOUT = "OUI' ,
◊PREFIXE = pref,
♦INDEX = (DM, Fm, dn, fn),

These key words make it possible to re-elect all the nodes of the mesh. The convention of renaming is the following one (in pseudonym FORTRAN):

no_fin (K8) = pref//no_mail (DM: Fm) //no_ini (dn: fn)

What wants to say that the name of a node will be formed while concaténant:

- the prefix possibly given by the user,
- a under-chain of characters extracted the name of the mesh which carries this node (cf convention C1 of elimination of the nodes stated above [§ 3]). One takes the characters of row ranging between DM and Fm. If DM > Fm, this under - character string is empty,
- a under-chain of characters extracted the name of the node (in its initial mesh). One takes the characters of row ranging between dn and fn. If dn > fn, this under-character string is empty.

It is necessary thus that: ltot = length (prefix) + (FM-dm+1) + (fn-dn+1) □ 8

One recalls that 2 nodes cannot have the same name in the same mesh. The goal of "clearance" for the user is to manage to re-elect some nodes (without too many forces of its share) in a conventional way without this convention leading to identical names.

A frequent case is the following:

if the meshes which gave rise to the macro-elements come from a pre - processor which generates names of nodes of the NOijklmn form and if the user gives to his super-meshes names with 2 characters: SA, SB,…. the sequence:

```
DEFI_NOEUD = _F (TOUT= "OUI', INDEX= (1, 2,3,8,))
```

will generate nodes of names: SA000001, SA000002,… , SB000001,

4.4.2 Operands NOEUD_FIN / SUPER_MAILLE / NOEUD_INIT

```
| ♦NOEUD_FIN        =no_fin      ,
  ♦SUPER_MAILLE =mail     ,
  ♦NOEUD_INIT       =no_ini      ,
```

These keywords make it possible to re-elect nodes **one by one**:

- no_fin is the name which one wants to give to the node mesh that one creates (final).
- mail and no_ini identify the node to be re-elected: mail is the name super - mesh which carries the node, no_ini is the name of the node in the mesh which was used to create the macro-element which defined the super - mesh mail.

4.5 Key word DEFI_GROUP_NO

◊DEFI_GROUP_NO =

This paragraph is almost identical to precedent (DEFI_NOEUD) by replacing key NOEUD_ by key GROUP_NO.

This key word factor makes it possible to define nodes groups from groups existing in the initial meshes of the macro-elements.

Note:

An initial nodes group can contain nodes which do not belong to edges of the macro-elements. These internal nodes thus do not exist in the final mesh. By convenience, one takes convention nevertheless to create the group reduced to his intersection with edge of the macro-element.

4.5.1 Operands TOUT / SUPER_MAILLE / PREFIXE / INDEX

```
|  ♦/TOUT=              "OUI',
     /SUPER_MAILLE = mail,
   ◊PREFIXE   = pref,
   ♦INDEX    = (DM, Fm, dn, fn),
```

These key keys make it possible to create all the nodes groups corresponding to the groups of the initial mesh associated with the mesh mail or all meshes if:

TOUT= "OUI'.

The convention of renaming is the following one (in pseudonym FORTRAN):

gno_fin (k8) = pref//no_mail (DM: Fm) //gno_ini (dn: fn)

What wants to say that the name of a nodes group will be formed while concaténant:

- the prefix possibly given by the user,
- a under-chain of characters extracted the name of the mesh,
- a under-chain of characters extracted the name of the group_no of the initial mesh.

It is necessary thus that:

ltot= length (prefix) + (FM-dm+1) + (fn-dn+1) □ 8

frequent cases is the following: the meshes which gave rise to the macro - elements come from a preprocessor which generates names of the GRNOijkl form. If the user gives to his super-meshes names with 2 characters: SA, SB,…., the sequence:

```
DEFI_GROUP_NO=_F (TOUT= "OUI', PREFIXE=' GN', INDEX=
(1,2,5,8))
```

Will generate nodes groups of names:

GNSA0001, GNSA0002,… , GNSB0001.

4.5.2 Operands GROUP_NO_FIN / SUPER_MAILLE / GROUP_NO_INIT

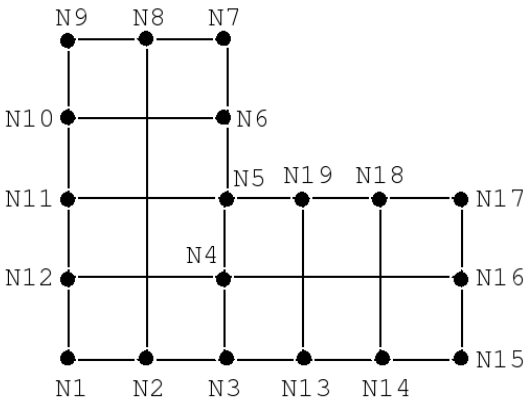
```
|  ♦GROUP_NO_FIN =gno_fin ,
   ♦SUPER_MAILLE  =mail   ,
   ♦GROUP_NO_INIT    =gno_ini ,
```

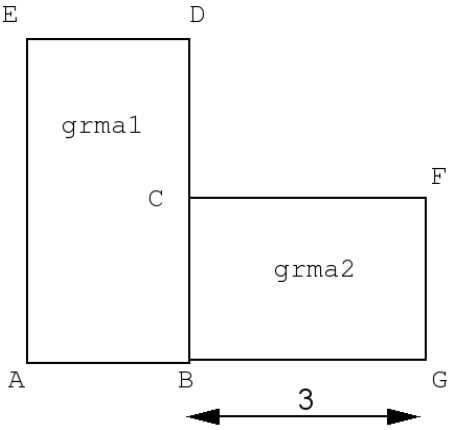
These keywords make it possible to create nodes groups **one by one** :

- gno_fin is the name which one wants to give to the GROUP_NO,
- mail and gno_ini identify the GROUP_NO initial:
  - mail is the name of the super-mesh which carries the GROUP_NO,
  - gno_ini is the name of the GROUP_NO of the initial mesh.

5 Example

Is the mesh m1:



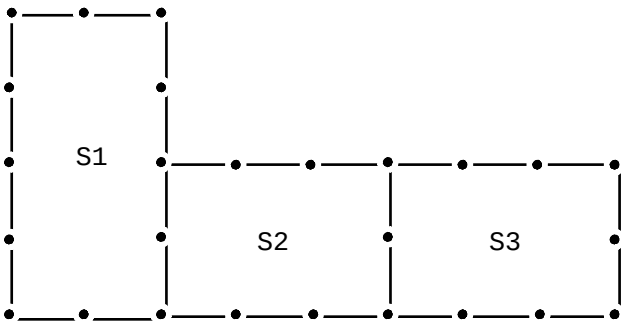



```
GROUP_NO:                          GROUP_MA:

AB = (N1 N2 N3)                     grma1
BC = (N3 N4 N5)                     grma2
CD = .....
......
```

On this mesh m1 one defines 2 macr_elem_stat.

```
mo1 =AFFE_MODELE          ( AFFE = _F (GROUP_MA = grma1)…)
mo2 =AFFE_MODELE          ( AFFE = _F (GROUP_MA = grma2)…)

S1 =MACR_ELEM_STAT        ( DEFINITION = _F   (= mo1 MODELS…)
                                EXTERIEUR  = _F   (GROUP_NO = (AB, BC, CD, OF,
EA))
                            …)
S2 =MACR_ELEM_STAT        ( DEFINITION = _F   (= mo2 MODELS…)
                                EXTERIEUR  = _F   (GROUP_NO = (BC, BG, FG,
CF))
                            …)
```

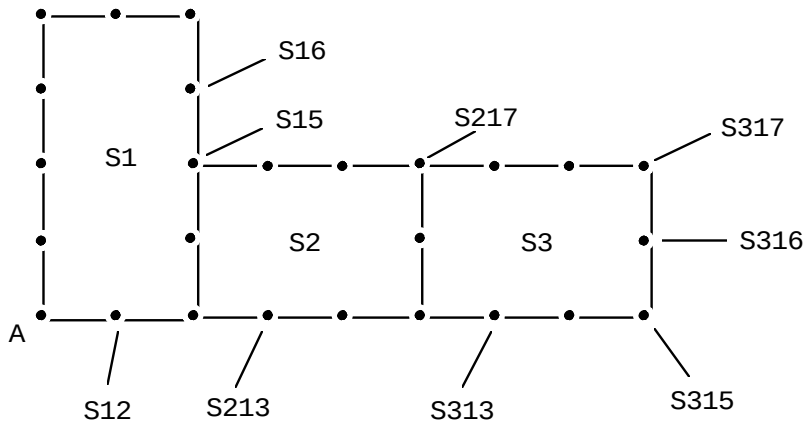
One can then define the mesh m2 :



```
m2 =DEFI_MAILLAGE       (

    DEFI_SUPER_MAILLE= (
        _F (MACR_ELEM = S1) ,
        _F (MACR_ELEM = S2  ,    SUPER_MAILLE = S2, ),
        _F (MACR_ELEM = S2  ,    SUPER_MAILLE = S3,  TRAN = 3.),),

    RECO_SUPER_MAILLE= (
        _F (SUPER_MAILLE= (S1, S2), GROUP_NO= (BC, BC), OPTION='
    NOEUD_A_NOEUD'),
        _F (SUPER_MAILLE= (S2, S3), GROUP_NO= (FG, BC), OPTION=' INVERSE'),),

    DEFI_NOEUD= (
        _F (TOUT = "OUI', INDEX = (1, 2,2,3)),
        _F (NOEUD_FIN = A,   SUPER_MAILLE = S1,  NOEUD_INIT = N1),),

    DEFI_GROUP_NO =
       _F (GROUP_NO_FIN = FG, SUPER_MAILLE = S3,    GROUP_NO_INIT = FG),)
```

The mesh obtained contains:

- 3 super-meshes : S1, S2, S3
- 26 nodes : A, S12,…, S317
- 1 GROUP_NO : FG = (S315, S316, S317)



Note::

The resticking of super-meshes could have been made more simply by:
RECO_GLOBAL = _F (TOUT = "OUI') .