

**Manuel d'Utilisation**  
**Fascicule U4.2- : Maillage**  
**Document : U4.23.03**

## Opérateur ASSE\_MAILLAGE

---

### 1 But

---

Assembler 2 maillages pour en former un nouveau.

Les deux maillages à assembler peuvent provenir des opérateurs : LIRE\_MAILLAGE [U4.21.01], DEFI\_MAILLAGE [U4.23.01], ASSE\_MAILLAGE [U4.23.03], ...

C'est-à-dire qu'ils peuvent contenir des mailles ordinaires et/ou des super-mailles porteuses de macro-éléments.

Produit une structure de données de type `maillage`.

---

## 2    Syntaxe

---

```
mac (maillage) =    ASSE_MAILLAGE (

%    noms des 2 maillages à assembler :
♦    MAILLAGE_1 = ma1 ,                    [maillage]
♦    MAILLAGE_2 = ma2 ,                    [maillage]

%    pour "recoller" les 2 maillages
/    OPERATION =   'COLLAGE',
     ♦    COLLAGE=_F(
         ♦    GROUP_MA_1 = gma1,   [gr_maille]
         ♦    GROUP_MA_2 = gma2,   [gr_maille] ),

%    pour "superposer" les 2 maillages
/    OPERATION =   'SUPERPOSE',

%    pour assembler des maillages contenant des
%    macro-éléments (sous-structures)
/    OPERATION =   'SOUS_STR',

)
```

---

### 3      Opérandes **MAILLAGE\_1** et **MAILLAGE\_2**

---

- ♦    **MAILLAGE\_1** = *ma1* , **MAILLAGE\_2** = *ma2* ,  
     *ma1* et *ma2* sont les noms des 2 maillages à assembler.

---

### 4      Opération = ' **SUPERPOSE** '

---

Avec ce choix : ' **SUPERPOSE** ', toutes les entités (mailles, nœuds, *group\_ma* et *group\_no*) des 2 maillages sont conservées. Le seul problème à régler concerne le nom des ces entités (conflit de noms possible car les 2 maillages peuvent contenir, par exemple, des nœuds de même nom).

Pour résoudre ces éventuels problèmes de noms :

- Les nœuds et les mailles sont toujours renommés. Cela veut dire que l'utilisateur ne peut pas connaître le nom des nœuds et mailles du maillage résultat. Il lui faudra en général utiliser (ce qui est toujours conseillé) les noms de *group\_no* et de *group\_ma*. Ou bien il devra imprimer le maillage pour connaître les noms choisis.
- Les noms des *group\_no* et *group\_ma* sont conservés dans la limite du possible. Si 2 *group\_ma* (ou 2 *group\_no*) ont le même nom dans les 2 maillages, le groupe venant du 2eme maillage est renommé automatiquement et le changement de nom est indiqué dans le fichier de message.

---

### 5      Opération = ' **COLLAGE** '

---

L'opération ' **COLLAGE** ' sert à connecter 2 maillages qui auraient été maillés indépendamment dans 2 fichiers distincts (par exemple par 2 équipes différentes).

Si les 2 maillages ne sont pas cohérents sur leur interface (discrétisation différente), l'utilisateur n'aura pas d'autre choix que d'utiliser la fonctionnalité *AFFE\_CHAR\_XXX/LIAISON\_MAILLE* pour "lier" les degrés de liberté des 2 maillages qui resteront topologiquement disjoints. Il devra alors utiliser l'opération ' **SUPERPOSE** '.

Si en revanche, les 2 maillages ont été prévus pour être recollés, il utilisera l'opération ' **COLLAGE** '. Pour cela, il aura de prendre la précaution de nommer les 2 groupes de mailles (d'interface) qui permettront de recoller les 2 maillages. De plus ces 2 groupes de mailles doivent être géométriquement coïncidents. L'utilisateur écrira alors :

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

Les *group\_ma* *gma1* et *gma2* seront alors fusionnés. Plus précisément :

- les mailles de *gma1* et *gma2* seront supprimées
- les nœuds de *gma2* subsisteront (mais seront orphelins)
- les nœuds de *gma1* connecteront les 2 maillages.

Avant de "fusionner" les 2 *group\_ma* *gma1* et *gma2*, le programme vérifiera que les nœuds de ces 2 *group\_ma* sont bien coïncidents (avec une tolérance de 0.001 fois la longueur de la plus petite arête des 2 maillages).

Les conflits de noms des entités des 2 maillages sont réglés de la même façon que pour l'opération ' **SUPERPOSE** ' (voir ci-dessus).

## 6      Opération = 'SOUS\_STR'

Pour mélanger dans un même modèle des éléments finis ordinaires et des macro-éléments (ou sous-structures), il faut disposer d'un maillage contenant à la fois des mailles ordinaires et des (super)mailles. L'opérateur `ASSE_CHAMP / OPERATION = 'SOUS_STR'` permet de constituer ce maillage "mixte" en assemblant un maillage ordinaire (ou mixte) et un maillage contenant des (super)mailles (provenant de `DEFI_MALLAGE`).

Le sens de l'assemblage est le suivant :

- **Toutes** les entités des 2 maillages arguments `ma1` et `ma2` (mailles, super-mailles, nœuds, groupe de mailles et groupe de nœuds) sont recopiées dans le maillage résultat : `mac`.
- Les seules entités partagées sont les **nœuds de mêmes noms**. Ce sont ces nœuds qui permettent d'assembler les 2 maillages.
- Traitement des entités portant le même nom :
  - nœuds : les nœuds du deuxième maillage portant un nom existant dans le premier maillage, ne sont pas ajoutés : on suppose que ce sont les mêmes. Les coordonnées du nœud conservé sont celles du nœud du premier maillage. Une alarme est émise lorsque la distance entre les deux nœuds confondus est supérieure à :

$$10^{-6} \cdot d\_refe$$

où  $d\_refe$  est une longueur caractéristique du maillage :

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

où  $d (O, N)$  est la distance du nœud  $N$  à l'origine du repère global.

- mailles (ou super-mailles) : si le maillage `ma2` contient des mailles de même nom que des mailles du premier maillage `ma1`, le programme s'arrête en erreur fatale.
- groupe de mailles (ou groupes de nœuds) : si le maillage `ma2` contient un groupe de mailles de même nom qu'un groupe de maille de `ma1`, celui-ci est ignoré et le programme émet un message d'alarme.

## 7      Exemple

### 7.1    Sous-structuration

On veut créer un maillage (`matot`) contenant des sous-structures statiques et un maillage ordinaire.

```
ma1      =  LIRE_MALLAGE ( )
macro1   =  MACR_ELEM_STAT ( ... )
macro2   =  MACR_ELEM_STAT ( ... )
masss    =  DEFI_MALLAGE (DEFI_MAILLE = _F (MACR_ELEM_STAT = macro1 ...))
matot    =  ASSE_MALLAGE (OPERATION='SOUS_STR',
                        MAILLAGE_1 = ma1, MAILLAGE_2 = masss)
```