Opérateurs AFFE_CHAR_THER et AFFE_CHAR_THER_F

1 But

Affecter des chargements et des conditions aux limites thermiques sur un modèle.

Pour l'opérateur AFFE_CHAR_THER, les valeurs affectées ne dépendent d'aucun paramètre et sont définies par des valeurs réelles.

Pour l'opérateur AFFE_CHAR_THER_F, les valeurs sont des fonctions d'un ou plusieurs paramètres à choisir dans l'ensemble (INST, X, Y, Z). Pour les chargements utilisés en thermique non-linéaire, les fonctions dépendent uniquement de la température (TEMP) et doivent être tabulées. De plus, pour certains chargements (par exemple SOURCE), les valeurs peuvent dépendre des variables de commande.

Ces fonctions doivent préalablement être définies par l'appel à un des opérateurs :

- DEFI_CONSTANTE [U4.31.01] ;
- DEFI_FONCTION [U4.31.02] ;
- DEFI_NAPPE [U4.31.03] ;
- FORMULE [U4.31.0 5] ;
- CALC_FONC_INTERP [U4.32.01].

Le concept produit est de type char_ther.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

Table des Matières

1But....1
2Syntaxe générale....5
3Généralités....6
4Opérandes....7
4.1Généralités sur les opérandes....7
4.1.1Les deux formes d'opérandes sous un mot clé facteur....7
4.1.2Entités topologiques d'affectation des chargements....7
4.2Opérande MODELE....7
4.3Opérande DOUBLE_LAGRANGE....7
4.4Mot-clé TEMP_IMPO....8
4.4.1But....8
4.4.2Syntaxe....8
4.4.3Opérandes....8
4.5Mot clé FLUX_REP....10
4.5.1But....10
4.5.2Syntaxe....10
4.5.3Opérandes....10
4.5.4Remarque....11
4.6Mot-clé FLUX_NL....13
4.6.1But....13
4.6.2Syntaxe....13
4.6.3Opérandes....13
4.7Mot-clé RAYONNEMENT....14
4.7.1But....14
4.7.2Syntaxe....14
4.7.3Opérandes....14
4.8Mot-clé ECHANGE....15
4.8.1But....15
4.8.2Syntaxe....15
4.8.3Opérandes....16
4.9Mot-clé SOURCE....17
4.9.1But....17
4.9.2Syntaxe....17
4.9.3Opérandes....17
4.10Mot-clé PRE_GRAD_TEMP....18
4.10.1But....18
4.10.2Syntaxe....18
4.10.3Opérandes....18

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

4.11Mot-clé LIAISON_DDL....20
4.11.1But....20
4.11.2Syntaxe....20
4.11.3Opérandes....20
4.12Mot-clé LIAISON_GROUP....22
4.12.1But....22
4.12.2Syntaxe....22
4.12.3Opérandes....23
4.12.4Utilisation de LIAISON_GROUP....25
4.13Mot-clé LIAISON_MAIL....27
4.13.1But....27
4.13.2Syntaxe....27
4.13.3Opérandes....27
4.13.3.1GROUP_MA_ESCL / GROUP_NO_ESCL....27
4.13.3.2GROUP_MA_MAIT....27
4.13.3.3 DISTANCE_MAX....28
4.13.3.4 DISTANCE_ALARME = d_ala....28
4.13.3.5CENTRE / ANGL_NAUT / TRAN....28
4.14Mot-clé ECHANGE_PAROI....28
4.14.1But....28
4.14.2Syntaxe....28
4.14.3Opérandes....30
4.14.3.1Cas de parois maillées....30
4.14.3.2Cas d'une ou plusieurs fissures X-FEM....31
4.14.4Utilisation de ECHANGE_PAROI....31
4.14.1Mailles et modélisations supportant ce chargement :....31
4.15Mot-clé LIAISON_UNIF....33
4.15.1But....33
4.15.2Syntaxe....33
4.15.3Opérandes....33
4.16Mot-clé LIAISON_CHAMNO....34
4.16.1But....34
4.16.2Syntaxe....34
4.16.3Opérandes....34
4.17Mot-clé CONVECTION....35
4.17.1But....35
4.17.2Syntaxe....35
4.17.3Opérande....35
4.18Mot-clé SOUR_NL....36
4.18.1But....36

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

4.18.2Syntaxe....36
4.18.3Opérandes....36
4.19Mot-clé EVOL_CHAR....36

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

2 Syntaxe générale

```
ch [char_ther]  =  AFFE_CHAR_THER

          (  ♦  MODELE =  mo,                                          [modele]
             ◊  DOUBLE_LAGRANGE  =  /'OUI',                            [défaut]
                                    /'NON'

             ♦  |  TEMP_IMPO =      (voir mot-clé TEMP_IMPO       )
                |  FLUX_REP =       (voir mot-clé FLUX_REP        )
                |  RAYONNEMENT =    (voir mot-clé RAYONNEMENT     )
                |  ECHANGE =        (voir mot-clé ECHANGE         )
                |  SOURCE =         (voir mot-clé SOURCE          )
                |  PRE_GRAD_TEMP =  (voir mot-clé PRE_GRAD_TEMP   )
                |  LIAISON_DDL =    (voir mot-clé LIAISON_DDL     )
                |  LIAISON_GROUP =  (voir mot-clé LIAISON_GROUP   )
                |  LIAISON_MAIL =   (voir mot-clé LIAISON_MAIL    )
                |  ECHANGE_PAROI =  (voir mot-clé ECHANGE_PAROI   )
                |  LIAISON_UNIF =   (voir mot-clé LIAISON_UNIF    )
                |  LIAISON_CHAMNO=  (voir mot-clé LIAISON_CHAMNO  )
                |  CONVECTION=      (voir mot-clé CONVECTION      )
                |  EVOL_CHAR=       (voir mot-clé EVOL_CHAR       )
           )


ch [char_ther]  =  AFFE_CHAR_THER_F

          (  ♦  MODELE =  mo,                                          [modele]
             ◊  DOUBLE_LAGRANGE  =  /'OUI',                            [défaut]
                                    /'NON'

             ♦  |  TEMP_IMPO =      (voir mot-clé TEMP_IMPO       )
                |  FLUX_REP =       (voir mot-clé FLUX_REP        )
                |  FLUX_NL =        (voir mot-clé FLUX_NL         )
                |  RAYONNEMENT =    (voir mot-clé RAYONNEMENT     )
                |  ECHANGE =        (voir mot-clé ECHANGE         )
                |  SOURCE =         (voir mot-clé SOURCE          )
                |  PRE_GRAD_TEMP=   (voir mot-clé PRE_GRAD_TEMP   )
                |  LIAISON_DDL =    (voir mot-clé LIAISON_DDL     )
                |  LIAISON_GROUP =  (voir mot-clé LIAISON_GROUP   )
                |  ECHANGE_PAROI =  (voir mot-clé ECHANGE_PAROI   )
                |  LIAISON_UNIF =   (voir mot-clé LIAISON_UNIF    )
                |  CONVECTION =     (voir mot-clé CONVECTION      )
                |  SOUR_NL =        (voir mot-clé SOUR_NL         )
           )
```


Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

3 Généralités

Messages d'erreur possibles liés à la commande AFFE_CHAR_THER

Il arrive parfois qu'une commande de calcul thermique (THER_LINEAIRE, THER_NON_LINE, ...) s'arrête en erreur fatale lors du calcul des seconds membres élémentaires dus aux chargements définis dans les commandes AFFE_CHAR_THER_*.

Lorsque le code s'arrête pendant ces calculs élémentaires, une information importante du message d'erreur est le nom de l'option de calcul demandée par le code. Le nom de cette option est en général inconnu de l'utilisateur et il lui est donc difficile de comprendre le message.

Dans le tableau ci-dessous, on donne en vis-à-vis des noms des options de calcul, le nom de la commande et du mot clé facteur qui permettent d'activer cette option.

Option de calcul élémentaire	Commande	Mot clé facteur
CHAR_THER_FLUNL	AFFE_CHAR_THER_F	FLUX_NL
CHAR_THER_FLUN_F	AFFE_CHAR_THER_F	FLUX_REP
CHAR_THER_FLUN_R	AFFE_CHAR_THER	FLUX_REP
CHAR_THER_FLUTNL	AFFE_CHAR_THER	CONVECTION
CHAR_THER_FLUTNL	AFFE_CHAR_THER_F	CONVECTION
CHAR_THER_FLUX_F	AFFE_CHAR_THER_F	FLUX_REP
CHAR_THER_FLUX_R	AFFE_CHAR_THER	FLUX_REP
CHAR_THER_GRAI_F	AFFE_CHAR_THER_F	PRE_GRAD_TEMP
CHAR_THER_GRAI_R	AFFE_CHAR_THER	PRE_GRAD_TEMP
CHAR_THER_PARO_F	AFFE_CHAR_THER_F	ECHANGE_PAROI
CHAR_THER_PARO_R	AFFE_CHAR_THER	ECHANGE_PAROI
CHAR_THER_SOUR_F	AFFE_CHAR_THER_F	SOURCE
CHAR_THER_SOUR_R	AFFE_CHAR_THER	SOURCE
CHAR_THER_TEXT_F	AFFE_CHAR_THER_F	ECHANGE
CHAR_THER_TEXT_R	AFFE_CHAR_THER	ECHANGE
CHAR_THER_SOURNL	AFFE_CHAR_THER_F	SOUR_NL

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

4 Opérandes

4.1 Généralités sur les opérandes

4.1.1 Les deux formes d'opérandes sous un mot clé facteur

Les opérandes sous un mot clé facteur sont de deux formes :

- les opérandes spécifiant les entités topologiques où sont affectés les chargements (mots-clés GROUP_NO et GROUP_MA, etc ...). Les arguments de ces opérandes sont identiques pour les deux opérateurs.
- le ou les opérandes spécifiant les valeurs affectées (TEMP, COEF_H, etc ...). La signification de ces opérandes est la même pour les deux opérateurs mais les arguments de ces opérandes sont tous du type réel pour l'opérateur AFFE_CHAR_THER et du type fonction (créé par l'un des opérateurs DEFI_FONCTION, DEFI_NAPPE, DEFI_CONSTANTE, FORMULE ou CALC_FONC_INTERP) pour l'opérateur AFFE_CHAR_THER _F.

Nous ne distinguerons donc pas dans ce document, sauf mention expresse du contraire, les deux opérateurs AFFE_CHAR_THER et AFFE_CHAR_THER_F.

4.1.2 Entités topologiques d'affectation des chargements

De façon générale, les entités topologiques sur lesquelles des valeurs doivent être affectées sont définies :

- par des nœuds en utilisant l'opérande GROUP_NO permettant d'introduire une liste de groupe de nœuds,

- par maille en utilisant l'opérande GROUP_MA permettant d'introduire une liste de groupes de mailles,

Règle :

Pour définir le domaine d'affectation le plus simplement possible, on utilise la règle de surcharge, c'est la dernière affectation qui prime.

4.2 Opérande MODELE

♦ MODELE = mo,

Concept produit par l'opérateur AFFE_MODELE [U4.41.01] où sont définis les types d'éléments finis affectés sur le maillage.

4.3 Opérande DOUBLE_LAGRANGE

◊ DOUBLE_LAGRANGE = 'OUI'/'NON'

Ce mot-clé permet de dire si l'utilisateur souhaite ou non dédoubler les multiplicateurs de Lagrange utilisés pour définir dualiser les conditions aux limites dans la matrice assemblée.
Concrètement, dédoubler les multiplicateurs de Lagrange permet d'utiliser des solveurs linéaires ne permettant pas le pivotage. Ne pas dédoubler les Lagrange permet de réduire le nombre de degré de liberté du problème (et donc la taille du problème à résoudre) mais son usage est limité aux solveurs MUMPS et Petsc.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

4.4 Mot-clé TEMP_IMPO

4.4.1 But

Mot clé facteur utilisable pour imposer, sur des nœuds ou des groupes de nœuds, une température.

Suivant le nom de l'opérateur appelé, les valeurs sont fournies directement (AFFE_CHAR_THER) ou par l'intermédiaire d'un concept de type fonction (AFFE_CHAR_THER_F).

4.4.2 Syntaxe

- pour AFFE_CHAR_THER

```
TEMP_IMPO = _F( ♦  |  TOUT     =       'OUI',
                   |  GROUP_NO =       lgno,              [l_gr_noeud]
                   |  GROUP_MA =       lgma,              [l_gr_maille]

                ♦  /  TEMP        =    t,                 [R]
                   /  |  TEMP_MIL =    tinf,              [R]
                      |  TEMP_INF =    tinf,              [R]
                      |  TEMP_SUP =    tsup,              [R]
              )
```

- pour AFFE_CHAR_THER_F

```
TEMP_IMPO = _F( ♦  |  TOUT     =       'OUI',
                   |  GROUP_NO =       lgno,              [l_gr_noeud]
                   |  GROUP_MA =       lgma,              [l_gr_maille]

                ♦  /  TEMP         =   tf,                [fonction]
                   /  |  TEMP_MIL  =   tf,                [fonction]
                      |  TEMP_INF  =   tinf,              [fonction]
                      |  TEMP_SUP  =   tsupf,             [fonction]

              )
```

4.4.3 Opérandes

/ TEMP =

Valeur de la **température** imposée sur le(s) nœud(s) spécifié(s).

/ Pour les éléments de coque thermique uniquement (Modélisation : 'COQUE') :

| TEMP_MIL

Température sur le feuillet moyen imposée sur le(s) nœud(s) spécifié(s).

| TEMP_INF

Température imposée sur la paroi inférieure de la coque.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

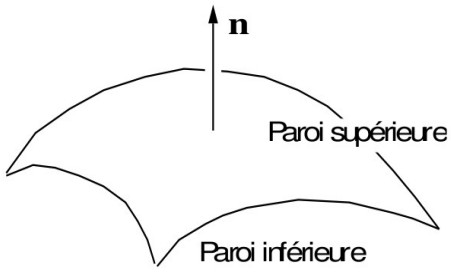
| TEMP_SUP

Température imposée sur la paroi supérieure de la coque.

Ces options permettent de représenter une variation parabolique de la température dans l'épaisseur.

Remarque :

La coque est orientée par la connectivité des nœuds de la maille associée (cf. [U3.01.00]). Soit n le vecteur normal orientant la coque :



Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

4.5 Mot clé FLUX_REP

4.5.1 But

Mot clé facteur utilisable pour appliquer des **flux normaux**, à une **face** d'élément volumique ou de coque thermique définie par une ou plusieurs mailles de type **triangle** ou **quadrangle**. Ce mot clé permet également d'appliquer un flux normal à une arête (en 2D PLAN ou AXIS ou AXIS_FOURIER) sur des mailles de type segment.

Suivant le nom de l'opérateur appelé, les valeurs sont fournies directement (AFFE_CHAR_THER) ou par l'intermédiaire d'un concept de type fonction (AFFE_CHAR_THER_F).

4.5.2 Syntaxe

- pour AFFE_CHAR_THER

```
 FLUX_REP = _F   (    ♦ /  TOUT      =       'OUI',
                        /  GROUP_MA =       lgma,            [l_gr_maille]

                     ♦  /  FLUN        =         fl,              [R]
                        /  |  FLUN_INF =       flin,            [R]
                           |  FLUN_SUP =       flsup,           [R]
                 )
```

- pour AFFE_CHAR_THER_F

```
FLUX_REP = _F (
                   ♦  /  TOUT      =       'OUI',
                      /  GROUP_MA =       lgma,            [l_gr_maille]


                   ♦  /  FLUN =              flf,             [fonction]
                      /  |  FLUN_INF =       flinf,           [fonction]
                         |  FLUN_SUP =       flsupf,          [fonction]
                      /  |  FLUX_X   =       flx,             [fonction]
                         |  FLUX_Y   =       fly,             [fonction]
                         |  FLUX_Z   =       flz,             [fonction]
                 )
```

4.5.3 Opérandes

/ FLUN : fl flux imposé normal à la maille.

Ce chargement s'applique aux types de mailles et aux modélisations suivantes :

Maille	Modélisation
TRIA3, TRIA6, QUAD4, QUAD8, QUAD9	3D, 3D_DIAG
SEG2, SEG3	PLAN, AXIS, AXIS_FOURIER, PLAN_DIAG, AXIS_DIAG

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

Plus précisément la condition aux limites appliquée est : $\lambda(grad\,T\cdot\boldsymbol{n})$ = fl

où λ est la conductivité thermique et $\boldsymbol{n}$ est la normale orientée dans le sens de la numérotation des nœuds de la maille. La convention d'orientation est celle utilisée dans AFFE_CHAR_MECA [U4.44.01].

```
/   |   FLUN_INF = flin
    |   FLUN_SUP = flsup
```

Flux normal imposé sur les parois inférieure et supérieure d'une coque thermique.

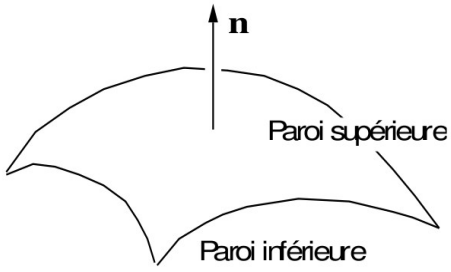
Ces chargements s'appliquent aux types de mailles et aux modélisations suivantes :

Maille	Modélisation
TRIA3, TRIA6	COQUE

$\boldsymbol{n}$ étant la normale orientant la surface [U4.44.01], la condition aux limites appliquée est :

$\lambda(grad\,T\cdot\boldsymbol{n})$ = flin où flin est le flux normal imposé sur la paroi inférieure de la coque,

$\lambda(grad\,T\cdot\boldsymbol{n})$ = flsup où flsup est le flux normal imposé sur la paroi supérieure de la coque.




```
/   |   FLUX_X = flx
    |   FLUX_Y = fly
    |   FLUX_Z = flz
```

Flux vectoriel $\boldsymbol{fl}$ dans le repère global (uniquement pour AFFE_CHAR_THER_F) que l'on projette sur la normale à l'élément (pour la définition de la normale [U4.44.01]).

$$\lambda(grad\,T\cdot\boldsymbol{n})=\boldsymbol{fl}\cdot\boldsymbol{n}=flx.nx+fly.ny+flz.nz$$

Ce chargement s'applique aux types de mailles et aux modélisations :

Maille	Modélisation
TRIA3, TRIA6, QUAD4, QUAD8, QUAD9	3D, 3D_DIAG
SEG2, SEG3	PLAN PLAN_DIAG

Remarque : la règle de rémanence (voir U1.03.00) s'applique entre les différentes quantités que l'on peut affecter : FLUN, FLUN_INF, ... FLUX_Z.

4.5.4 Remarque

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

Le mot-clé simple CARA_TORSION de ce mot-clé facteur FLUX_REP n'est pas documenté ici et ne doit pas être employé par l'utilisateur. Il ne sert qu'à la macro-commande MACR_CARA_POUTRE. Celle-ci sert à identifier les caractéristiques géométriques des sections de poutres. Pour les caractéristiques de torsion, la commande résout un problème de laplacien en employant de façon indirecte les opérateurs de thermique linéaire.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

4.6 Mot-clé FLUX_NL

4.6.1 But

Mot-clé facteur utilisable pour appliquer des **flux normaux** fonctions de la température, à une **face** d'élément volumique définie par une ou plusieurs mailles de type **triangle** ou **quadrangle**. Ce mot clé permet également d'appliquer un flux normal à une arête (en 2D PLAN ou AXIS) sur des mailles de type segment. On peut ainsi modéliser une condition de rayonnement du type loi de Stefan.
Ce type de flux n'est disponible que dans la commande AFFE_CHAR_THER_F et n'est utilisé que par les commandes THER_NON_LINE [U4.54.02] et THER_NON_LINE_MO [U4.54.03].

Les valeurs sont fournies par un concept de type fonction. La fonction dépend de la température, à l'exclusion de tout autre paramètre. En outre, il s'agit nécessairement d'une fonction tabulée et non d'une formule.

4.6.2 Syntaxe

- Pour AFFE_CHAR_THER_F

```
FLUX_NL = _F (
                ♦  /  TOUT     =       'OUI',
                   /  GROUP_MA =       lgma,            [l_gr_maille]

                ♦     FLUN     =          fl,              [fonction]
              )
```

4.6.3 Opérandes

FLUN : flux imposé normal à la maille.

Ce chargement s'applique aux types de mailles et aux modélisations suivantes :

Maille	Modélisation
TRIA3, TRIA6, QUAD4, QUAD8, QUAD9	3D, 3D_DIAG
SEG2, SEG3	PLAN, AXIS PLAN_DIAG, AXIS_DIAG

Plus précisément la condition aux limites appliquée est :

$\lambda\left(grad\,T\cdot\boldsymbol{n}\right)$ = fl

où $\boldsymbol{n}$ est la normale orientée dans le sens de la numérotation des nœuds de la maille. Orientation utilisée dans AFFE_CHAR_MECA document [U4.44.01].

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

4.7 Mot-clé RAYONNEMENT

4.7.1 But

Mot-clé permettant de définir le flux rayonné à l'infini suivant la formule :

$$\Phi_{ray}=\sigma\,\epsilon\left([T+273,15]^4-[T_\infty+273,15]^4\right)$$

par la donnée de l'émissivité ε, la constante de Boltzmann σ et la température à l'infini T_∞ exprimée en Celsius. La température T sera elle aussi exprimée en Celsius, il faut donc veiller, par cohérence, à n'utiliser que des degrés Celsius pour toute l'étude.

4.7.2 Syntaxe

- pour AFFE_CHAR_THER

```
RAYONNEMENT = _F (
                      ♦  /  TOUT     =       'OUI',
                         /  GROUP_MA =       lgma,            [l_gr_maille]

                       ♦     SIGMA    =       sigma,           [R8]
                       ♦     EPSILON  =       epsilon,         [R8]
                       ♦     TEMP_EXT=        tex,             [R8]
                    )
```

- pour AFFE_CHAR_THER_F

```
 RAYONNEMENT = _F   (
                      ♦  /  TOUT     =       'OUI',
                         /  GROUP_MA =       lgma,            [l_gr_maille]

                      ♦     SIGMA    =       sigma,           [fonction]
                      ♦     EPSILON  =       epsilon,         [fonction]
                      ♦     TEMP_EXT=        tex,             [fonction]
                   )
```

4.7.3 Opérandes

♦ SIGMA = sigma
♦ EPSILON = epsilon
♦ TEMP_EXT = tex

Ce chargement s'applique aux modélisations suivantes :

Maille	Modélisation
TRIA3, TRIA6, QUAD4, QUAD8, QUAD9	3D, 3D_DIAG
SEG2, SEG3	PLAN, AXIS PLAN_DIAG, AXIS_DIAG

sigma : constante de Boltzmann, $\sigma=5.6710^{-8}$ en unités SI ($W/m^2.K^4$) (attention à cette valeur si les unités de maillage changent),
epsilon : émissivité,
tex : température à l'infini en degrés Celsius.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

4.8 Mot-clé ECHANGE

4.8.1 But

Mot-clé facteur utilisable pour appliquer des **conditions d'échange** avec une température extérieure à une **face** d'éléments volumiques ou de coques, définie par une ou plusieurs mailles de type **triangle** ou **quadrangle**. Ce mot clé permet également d'appliquer des conditions d'échange à une arête (en 2D PLAN ou AXIS ou AXIS_FOURIER) sur des mailles de type segment.

Suivant le nom de l'opérateur appelé, les valeurs sont fournies directement (AFFE_CHAR_THER) ou par l'intermédiaire d'un concept de type fonction (AFFE_CHAR_THER_F).

4.8.2 Syntaxe

- pour AFFE_CHAR_THER

```
ECHANGE = _F (
                  ♦  /  TOUT     =       'OUI',
                     /  GROUP_MA =       lgma,            [l_gr_maille]

                  ♦  /  ♦  COEF_H   =          h,           [R]
                        ♦  TEMP_EXT =          tex,         [R]

                     /  |  ♦  COEF_H_INF   =  hin,         [R]
                           ♦  TEMP_EXT_INF =  texin,       [R]
                        |  ♦  COEF_H_SUP   =  hsup,        [R]
                           ♦  TEMP_EXT_SUP =  texsup,      [R]
               )
```

- pour AFFE_CHAR_THER_F

```
ECHANGE = _F (
                  ♦  /  TOUT     =       'OUI',
                     /  GROUP_MA =       lgma,            [l_gr_maille]

                  ♦  /  ♦  COEF_H   =          hf,          [fonction]
                        ♦  TEMP_EXT =          texf,        [fonction]

                     /  |  ♦  COEF_H_INF   =  hinf,        [fonction]
                           ♦  TEMP_EXT_INF =  texinf,      [fonction]
                        |  ♦  COEF_H_SUP   =  hsupf,       [fonction]
                           ♦  TEMP_EXT_SUP =  texsupf,     [fonction]
               )
```


Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

4.8.3 Opérandes

/ ♦ COEF_H = h,
♦ TEMP_EXT = tex,

Ce chargement s'applique aux types de mailles et aux modélisations suivantes :

Maille	Modélisation
TRIA3, TRIA6, QUAD4, QUAD8, QUAD9	3D, 3D_DIAG
SEG2, SEG3	PLAN, PLAN_DIAG AXIS, AXIS_FOURIER, AXIS_DIAG

Plus précisément la condition aux limites appliquée est :

$$\lambda(grad\,T\cdot\boldsymbol{n})=h(tex-T)\quad(h>0)$$

où $\boldsymbol{n}$ est la normale orientée dans le sens de la numérotation des nœuds sommets (orientation utilisée dans AFFE_CHAR_MECA [U4.44.01]).

/ | ♦ COEF_H_INF = hin,
♦ TEMP_EXT_INF = texin,
| ♦ COEF_H_SUP = hsup,
♦ TEMP_EXT_SUP = texsup,

Ce chargement s'applique aux types de mailles et aux modélisations suivantes :

Maille	Modélisation
TRIA3, TRIA6	COQUE

$\boldsymbol{n}$ étant la normale orientant la surface [U2.03.03], la condition aux limites appliquée est :

$$\lambda(grad\,T\cdot\boldsymbol{n})=hin(texin-T)$$

où hin coefficient d'échange sur la paroi inférieure de la coque,
et texin température extérieure, coté paroi inférieure.

$$\lambda(grad\,T\cdot\boldsymbol{n})=hsup(texsup-T)$$

où hsup coefficient d'échange sur la paroi supérieure de la coque,
et texsup température extérieure, coté paroi extérieure.

Remarques :
• la règle de rémanence (voir U1.03.00) s'applique entre les différentes quantités que l'on peut affecter : COEF_H, COEF_H_INF, ... TEMP_EXT_SUP.
• Dans le cas des chargements fonctions (AFFE_CHAR_THER_F), comme il y a deux composantes distinctes (le coefficient d'échange et la température), qui sont *a priori* indépendantes, on ne peut pas utiliser simultanément ce chargement avec une fonction multiplicatrice dans l'opérateur de résolution (THER_LINEAIRE ou THER_NON_LINE). Un message d'erreur vous avertira.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

4.9 Mot-clé SOURCE

4.9.1 But

Mot-clé facteur utilisable pour appliquer des **sources volumiques** (2D ou 3D) à un **domaine** défini par une ou plusieurs mailles de type **volumique**.

Suivant le nom de l'opérateur appelé, les valeurs sont fournies directement (AFFE_CHAR_THER) ou par l'intermédiaire d'un concept de type fonction (AFFE_CHAR_THER_F). Ce chargement accepte des fonctions dépendant des variables de commande.

4.9.2 Syntaxe

- pour AFFE_CHAR_THER

```
SOURCE=_F (
                  ♦  /  TOUT     =       'OUI',
                     /  GROUP_MA =       lgma,            [l_gr_maille]

                   ♦..... / SOUR          =        s,     [R]
                     /  SOUR_CALCULEE =        chs,   [cham_elem]
          )
```

- pour AFFE_CHAR_THER_F

```
SOURCE=_F (
               ♦  /  TOUT     =       'OUI',
                  /  GROUP_MA =       lgma,            [l_gr_maille]

               ♦  SOUR        =          sf,           [fonction]
          )
```

4.9.3 Opérandes

Ce chargement s'applique aux types de mailles et aux modélisations suivantes :

Maille	Modélisation
HEX A8, HEXA20, HEXA27 PYRA5, PYRA13, PENTA6, PENTA15 TETRA4, TETRA10	3D, 3D_DIAG
TRIA3, TRIA6, QUAD4, QUAD8, QUAD9	PLAN, PLAN_DIAG, AXIS, AXIS_FOURIER AXIS_DIAG

/ ♦ SOUR = s,

Valeur de la source supposée constante sur l'élément.

/ ♦ SOUR_CALCULEE = chs,

Nom du cham_elem_sour_R contenant sur chaque élément les valeurs de la source discrétisée aux points de Gauss (1ère famille).

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

4.10 Mot-clé PRE_GRAD_TEMP

4.10.1 But

Mot-clé facteur utilisable pour appliquer à un élément 3D ou 2D (PLAN, AXIS) un gradient de température supposé uniforme dans l'élément. Ce gradient de température "initial" est utilisable par exemple pour résoudre les problèmes élémentaires déterminant les correcteurs de thermique linéaire stationnaire dans la cellule de base (2D, 3D), en homogénéisation périodique.

Les coefficients de conductibilité homogénéisée sont obtenus en calculant par l'opérateur POST_ELEM [U4.81.22] mot-clé ENER_POT l'énergie dissipée thermiquement à l'équilibre en thermique linéaire à partir des correcteurs.

A cause de l'analogie thermique, cette démarche peut être exploitée pour obtenir les correcteurs en élasticité antiplane dans la cellule de base 2D, aussi bien qu'en conduction électrique.

Le calcul utilise la conductivité du matériau. Celle-ci est supposée être isotrope et indépendante de la température.

L'affectation peut se faire sur une ou plusieurs mailles ou sur tous les éléments du modèle.

4.10.2 Syntaxe

- pour AFFE_CHAR_THER

```
PRE_GRAD_TEMP = _F  (
                  ♦  /  TOUT      =       'OUI',
                     /  GROUP_MA  =       lgma,            [l_gr_maille]

                  ♦  |  FLUX_X =          flx,                [R]
                     |  FLUX_Y =          fly,                [R]
                     |  FLUX_Z =          flz,                [R]
                     )
```

4.10.3 Opérandes

Ce chargement s'applique aux types de mailles et aux modélisations suivantes :

Maille	Modélisation
TRIA3, TRIA6, QUAD4, QUAD8, QUAD9	PLAN, AXIS, PLAN_DIAG, AXIS_DIAG
HEXA8, HEXA20, HEXA27 PENTA6, PENTA15, TETRA4, TETRA10 PYRA5, PYRA13	3D, 3D_DIAG

♦ | FLUX_X = flx (flxf)
| FLUX_Y = fly (flyf)
| FLUX_Z = flz (flzf) (en 3D seulement)

Composantes du gradient de température $grad\,T_{ini}$ dans le repère global.

Le second membre élémentaire calculé est : $\int_{V_e} grad\,T_{ini}\,K\,grad\,T^{*}\,dV_e$ où K est le tenseur des conductivités thermiques.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

Les gradients peuvent être fonction de la géométrie et/ou du temps.

- pour AFFE_CHAR_THER_F

```
PRE_GRAD_TEMP = _F  (
                    ♦  /  TOUT     =       'OUI',
                       /  GROUP_MA =       lgma,            [l_gr_maille]

                       ♦  |  FLUX_X =          flxf,         [fonction]
                          |  FLUX_Y =          flyf,         [fonction]
                          |  FLUX_Z =          flzf,         [fonction]
                    )
```


Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

4.11 Mot-clé LIAISON_DDL

4.11.1 But

Mot-clé facteur utilisable pour définir une relation linéaire entre des degrés de liberté de deux ou plusieurs nœuds.

Suivant le nom de l'opérateur appelé, les valeurs sont fournies directement (AFFE_CHAR_THER) ou par l'intermédiaire d'un concept fonction (AFFE_CHAR_THER_F).

4.11.2 Syntaxe

- pour AFFE_CHAR_THER

```
LIAISON_DDL = _F (♦ GROUP_NO =       lgno,            [l_gr_noeud]

                  ◊ DDL =      |            'TEMP',       [DEFAUT]
                               |            'TEMP_MIL',
                               |            'TEMP_INF',
                               |            'TEMP_SUP',
                               |            'H1',

                  ♦ COEF_MULT =             ai,          [l_R]
                  ♦ COEF_IMPO =             b ,          [R]
                )
```

- pour AFFE_CHAR_THER_F

```
LIAISON_DDL = _F (  ♦  GROUP_NO =       lgno,          [l_gr_noeud]

                  ◊ DDL =      |            'TEMP',       [DEFAUT]
                               |            'TEMP_MIL',
                               |            'TEMP_INF',
                               |            'TEMP_SUP',
                               |            'H1',

                  ♦  COEF_MULT =            ai ,         [l_R]
                  ♦  COEF_IMPO =            bf ,         [fonction]
                )
```

4.11.3 Opérandes

La liste des nœuds N_i ($i=1,r$) définie par GROUP_NO est ordonnée de façon naturelle, c'est-à-dire dans l'ordre de la liste de groupe de nœuds, et pour chaque groupe de nœuds, dans l'ordre de définition du groupe par GROUP_NO.

L'argument de DDL doit être une liste de degrés de liberté T_i ($i=1,r$) de r textes pris parmi :
'TEMP', 'TEMP_MIL', 'TEMP_SUP', 'TEMP_INF', 'H1'

Si le mot clé DDL est omis, par défaut la relation linéaire portera sur les degrés de liberté 'TEMP'.

L'argument de COEF_MULT doit être une liste a_i ($i=1,r$) de coefficients (de type réel pour AFFE_CHAR_THER et AFFE_CHAR_THER_F).

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

L'argument de COEF_IMPO est un coefficient β pour AFFE_CHAR_THER, une fonction de l'espace pour AFFE_CHAR_THER_F.

La condition cinématique suivante est appliquée : $\sum_{i=1}^{r} \alpha_i T_i = \beta$

Remarques :

*Les composantes 'TEMP_MIL', 'TEMP_SUP' et 'TEMP_INF' ne peuvent intervenir que dans des combinaisons affectées **uniquement** à des nœuds qui appartiennent à des éléments de **coque** (modélisation 'COQUE').*

*La composante 'H1' ne peut intervenir que dans des combinaisons affectées **uniquement** à des nœuds qui appartiennent à des éléments **X-FEM**. Dans ce cas, seuls les degrés de liberté 'TEMP' et 'H1' peuvent figurer dans la relation linéaire.*

*Dans le cas d'une relation linéaire entre les degrés de liberté d'un même nœud, on répétera derrière le mot clé GROUPE_NO le nom du groupe définissant le nœud autant de fois qu'il y a de degrés de liberté dans la relation. **Exemple :** pour imposer $T_{\text{sup}} = T_{\text{inf}}$ sur le nœud de groupe $GN1$, on écrira :*

```
LIAISON_DDL = _F (  GROUP_NO  =  (GN1,GN1),
                    DDL       =  ('TEMP_SUP', 'TEMP_INF'),
                    COEF_MULT =  (1.,-1.),
                    COEF_IMPO =  0.,
                 )
```

Il est important de noter qu'à une occurrence du mot clé facteur LIAISON_DDL correspond une et une seule relation linéaire.
*Si on veut imposer la même relation entre 2 groupes de nœuds GRN01 et GRN02 (même température nœud à nœud par exemple) **on ne peut pas écrire :***

```
LIAISON_DDL = _F    ( GROUP_NO  =  (GRN01, GRN02),
                      DDL       =  ('TEMP',  'TEMP'),
                    COEF_MULT =  (1.,-1.),
                    COEF_IMPO =  0.,                    )
```

Cette écriture n'a de sens que si GRN01 et GRN02 ne contiennent chacun qu'un seul nœud. Il faudra dans le cas ci-dessus expliciter chaque relation linéaire, nœud par nœud.

Le mot-clé LIAISON_GROUP permet par contre de condenser l'écriture des relations linéaires entre 2 groupes de nœuds en vis-à-vis.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

4.12 Mot-clé LIAISON_GROUP

4.12.1 But

Mot-clé facteur utilisable pour définir des relations linéaires entre des couples de nœuds, ces couples de nœuds étant obtenus en mettant en vis-à-vis deux listes de mailles ou de nœuds.

Suivant le nom de l'opérateur appelé N, les valeurs sont fournies directement (AFFE_CHAR_THER) ou par l'intermédiaire d'un concept fonction (AFFE_CHAR_THER_F).

4.12.2 Syntaxe

- pour AFFE_CHAR_THER

```
LIAISON_GROUP=_F (  ♦  /  ♦  GROUP_MA_1    =  lgma1, [l_gr_maille]
                          ♦  GROUP_MA_2    =  lgma2, [l_gr_maille]
                       /  ♦  GROUP_NO_1    =  lgno1, [l_gr_noeud]
                          ♦  GROUP_NO_2    =  lgno2, [l_gr_noeud]
                    ◊  SANS_GROUP_NO       =  lgno,  [l_gr_noeud]
                     ◊  DDL_1 =   |     'TEMP',          [DEFAUT]
                                  |     'TEMP_MIL',
                                  |     'TEMP_INF',
                                  |     'TEMP_SUP',
                                  |     'H1',
                     ◊  DDL_2 =   |     'TEMP',          [DEFAUT]
                                  |     'TEMP_MIL',
                                  |     'TEMP_INF',
                                  |     'TEMP_SUP',
                                  |     'H1',
                     ♦  COEF_MULT_1 =      a1i,         [l_R]
                     ♦  COEF_MULT_2 =      a2i,         [l_R]
                     ♦  COEF_IMPO       =  b   ,        [R]
                    ◊  |  CENTRE    =  lr ,             [l_R]
                       |  ANGL_NAUT =  lr ,             [l_R]
                       |  TRAN      =  lr ,             [l_R]
                    ◊  SOMMET       =  'OUI',
                )
```


Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

- pour AFFE_CHAR_THER_F

```
LIAISON_GROUP=_F (  ♦  /  ♦  GROUP_MA_1    =  lgma1, [l_gr_maille]
                          ♦  GROUP_MA_2    =  lgma2, [l_gr_maille]
                       /  ♦  GROUP_NO_1    =  lgno1, [l_gr_noeud]
                          ♦  GROUP_NO_2    =  lgno2, [l_gr_noeud]
                    ◊  SANS_GROUP_NO       =  lgno,  [l_gr_noeud]
                     ◊  DDL_1 =   |     'TEMP',          [DEFAUT]
                                  |     'TEMP_MIL',
                                  |     'TEMP_INF',
                                  |     'TEMP_SUP',
                                  |     'H1',
                     ◊  DDL_2 =   |     'TEMP',          [DEFAUT]
                                  |     'TEMP_MIL',
                                  |     'TEMP_INF',
                                  |     'TEMP_SUP',
                                  |     'H1',
                    ♦   COEF_MULT_1 =   a1i   ,          [l_R]
                    ♦   COEF_MULT_2 =   a2i   ,          [l_R]
                    ♦  COEF_IMPO    =   bf,              [fonction]
                    ◊  |  CENTRE    =  lr ,              [l_R]
                       |  ANGL_NAUT =  lr ,              [l_R]
                       |  TRAN      =  lr ,              [l_R]
                    ◊  SOMMET          =  'OUI'
                )
```

4.12.3 Opérandes

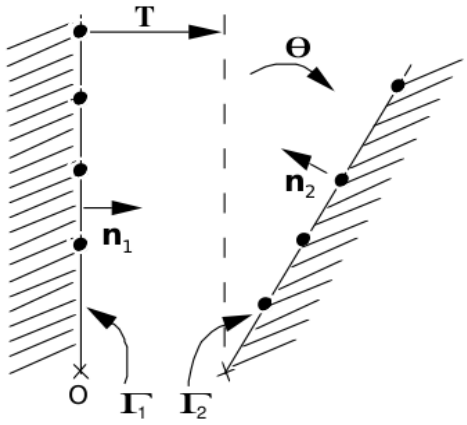


Figure 4.11.3-a : Transformation géométrique d'une frontière en une autre

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

Condition cinématique "générale" : $\sum_{i=1}^{NDDL1} \alpha_{1i} T_i|_{\Gamma_1} + \sum_{i=1}^{NDDL2} \alpha_{2i} T_i|_{\Gamma_2} = \beta$

/ ♦ GROUP_MA_1 =

Cette opérande définit Γ_1 par l'intermédiaire des mailles qui le composent.

♦ GROUP_MA_2 =

Cette opérande définit Γ_2 par l'intermédiaire des mailles qui le composent.

/ ♦ GROUP_NO_1 =

Cette opérande définit Γ_1 par l'intermédiaire des nœuds qui le composent.

♦ GROUP_NO_2 =

Cette opérande définit Γ_2 par l'intermédiaire des nœuds qui le composent.

◊ SANS_GROUP_NO =

Cette opérande permet de supprimer de la liste des couples de nœuds en vis-à-vis tous les couples dont au moins un des nœuds appartient à la liste de nœuds donnée.

Cela permet d'éviter l'accumulation de relations linéaires sur un même nœud au cours de différentes itérations sur le mot clé facteur LIAISON_GROUP ce qui conduit la plupart du temps à une matrice singulière.

♦ COEF_MULT_1 (resp. COEF_MULT_2)

Liste de réels exactement dimensionnée au nombre de degrés de liberté déclarés dans DDL_1 (resp. DDL_2) correspondant aux coefficients multiplicateurs de la relation linéaire.

♦ COEF_IMPO : coefficient de blocage de la relation linéaire :

β : réel pour AFFE_CHAR_THER
β_f : fonction pour AFFE_CHAR_THER_F

◊ CENTRE : coordonnées du centre de rotation
◊ ANGL_NAUT : angles nautiques en degrés définissant la rotation (voir AFFE_CARA_ELEM [U4.42.01] mot clé ORIENTATION)
◊ TRAN : composantes du vecteur translation

Ces opérandes permettent de définir une transformation virtuelle (rotation et/ou translation) approximative de Γ_1 en Γ_2 afin d'assurer la bijectivité de la fonction vis-à-vis.

◊ DDL_1 (resp. DDL_2) :

Liste de textes pris parmi :
'TEMP', 'TEMP_MIL', 'TEMP_INF', 'TEMP_SUP', 'H1'

'TEMP_MIL', 'TEMP_INF' et 'TEMP_SUP' ne peuvent être utilisés que pour des éléments de coque thermique (modélisation : 'COQUE').

'H1' ne peut être utilisé que pour des éléments X-FEM. Dans ce cas, seuls les degrés de liberté 'TEMP' et 'H1' peuvent figurer dans les relations linéaires.

Par défaut, le degré de liberté considéré pour tous les nœuds des relations linéaires est 'TEMP'.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

◊ SOMMET = 'OUI'

Lorsque les mailles de bord sont quadratiques, l'utilisation de SOMMET : 'OUI' force l'algorithme d'appariement à associer les nœuds sommets à d'autres nœuds sommets. Dans le cas de maillages fins, cela permet dans certains cas d'éviter les problèmes de conflits de vis-à-vis.

4.12.4 Utilisation de LIAISON_GROUP

- LIAISON_GROUP ne génère des relations linéaires qu'entre 2 nœuds (un sur Γ_1, un sur Γ_2)

  Pour générer des relations linéaires sur plus de 2 nœuds, utiliser le mot clé LIAISON_DDL.

- détermination des couples de nœuds en vis-à-vis :

  dans un premier temps, on établit les deux listes de nœuds à mettre en vis-à-vis (ie à apparier), pour chaque occurrence du mot-clé facteur LIAISON_GROUP :

  - pour les mots-clés GROUP_NO_1 et GROUP_NO_2, ce sont les nœuds constituant les groupes de nœuds,
  - pour les mots-clés GROUP_MA_1 et GROUP_MA_2, ce sont les nœuds des mailles constituant les groupes de mailles.

  Les redondances étant éliminées, les deux listes de nœuds obtenues doivent avoir la même longueur.

  La détermination des couples de nœuds en vis-à-vis se fait en plusieurs étapes :

  - pour chaque nœud $N1$ de la première liste, on cherche le nœud image $N2=f(N1)$ de la deuxième liste. Si f n'est pas injective (un nœud $N2$ est l'image de deux nœuds distincts $N1$ et $N1'$), le message d'erreur suivant est émis :

    ```
    <F> <AFFE_CHAR_THER> <PACOAP> CONFLIT DANS LES VIS-A-VIS
    DES NOEUDS
    LE NOEUD N2 EST LE VIS-A-VIS DES NOEUDS N1 ET N1'
    ```

  - pour chaque nœud $N2$ de la deuxième liste, on cherche le nœud image $N1=g(N2)$ de la première liste. Si g n'est pas injective (un nœud $N1$ est l'image de deux nœuds distincts $N2$ et $N2'$), le message d'erreur suivant est émis :

    ```
    <F> <AFFE_CHAR_MECA> <PACOAP> CONFLIT DANS LES VIS-A-VIS
    DES NOEUDS
    LE NOEUD N1 EST LE VIS-A-VIS DES NOEUDS N2 ET N2'
    ```

  - on vérifie que $g=f^{-1}$, c'est-à-dire que les couples obtenus par les étapes a) et b) sont les mêmes (on veut avoir une bijection f entre les deux listes de nœuds). Si f n'est pas surjective, le message d'erreur suivant est émis :

    ```
    <F> <AFFE_CHAR_MECA> <PACOAP> CONFLIT DANS LES VIS-A-VIS GENERES
    SUCCESSIVEMENT A PARTIR DES LISTES LIST1 ET LIST2
    LE NOEUD DE LA PREMIERE LISTE N1 N'EST L'IMAGE D'AUCUN NOEUD PAR
    LA CORRESPONDANCE INVERSE
    ```

  Pour un nœud N donné, on appelle nœud image $f(N)$ le nœud de l'autre liste de nœuds qui réalise le minimum de la distance avec N. Pour faciliter l'appariement, notamment dans le cas de géométries particulières (où les frontières Γ_1 et Γ_2 pourraient "presque" se déduire l'une de l'autre par la composition d'une translation et d'une rotation), on offre la possibilité de faire une transformation géométrique virtuelle du premier groupe de nœuds (translation et rotation (cf. [Figure 4.11.3-a]) avant de calculer les distances (mots-clés TRAN, CENTRE et ANGL_NAUT).

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

Pour chaque occurrence du mot-clé facteur LIAISON_GROUP, on construit ainsi la liste des nouveaux couples en vis-à-vis. Lorsqu'on a balayé toutes les occurrences, on supprime de la liste les couples en double.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

Remarque :

Dans les couples de nœuds en vis-à-vis, l'ordre des nœuds est important. Si pour la première occurrence de LIAISON_GROUP, un nœud N appartenait au premier groupe de nœuds et un nœud M au deuxième groupe de nœuds, et que pour la seconde occurrence de LIAISON_GROUP, c'est l'inverse, on obtiendra à l'issue de l'appariement les couples (N,M) et (M,N). Ils ne seront pas éliminés lors de la détection des redondances ; par contre, la matrice obtenue sera singulière. Ainsi, on conseille de garder la même logique lors de la description des bords en vis-à-vis.

4.13 Mot-clé LIAISON_MAIL

4.13.1 But

Mot-clé facteur permettant de "recoller thermiquement" deux bords d'une structure. Ces bords peuvent être maillés différemment (maillages incompatibles) mais doivent se déduire l'un de l'autre par rotation et/ou translation.

4.13.2 Syntaxe

- dans AFFE_CHAR_THER seulement

```
LIAISON_MAIL =_F (  ♦  GROUP_MA_MAIT =        lgma_mait,

                    ♦  |  GROUP_MA_ESCL =       lgma_escl,
                       |  GROUP_NO_ESCL =       lgno_escl,

                    ◊     DISTANCE_MAX   =     d_max,        [R]

                    ◊  |  ♦  TRAN   =     (tx, ty, [tz]),        [l_R]
                       |  ♦  CENTRE =     (xc, yc, [zc]),        [l_R]
                          ♦  ANGL_NAUT =  (alpha, [beta, gamma]),[l_R]
                  )
```

La face 1 est appelée face "maître" ; la face 2 est appelée face "esclave".

4.13.3 Opérandes

4.13.3.1 GROUP_MA_ESCL / GROUP_NO_ESCL

Ces mots-clés permettent de définir l'ensemble des nœuds de la face esclave. On prend tous les nœuds spécifiés par le mot-clé GROUP_NO_ESCL plus éventuellement les nœuds portés par les mailles spécifiées par le mot-clé GROUP_MA_ESCL.

4.13.3.2 GROUP_MA_MAIT

Ce mot-clé permet de définir l'ensemble des mailles où l'on cherchera les vis-à-vis des nœuds de la face esclave.

Il ne faut pas donner les mailles de surface (en 3D) composant la face maître, mais les mailles volumiques adjacentes à la face maître. Les mailles spécifiées sont des candidates pour la recherche des vis-à-vis. On peut en donner trop.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

4.13.3.3 DISTANCE_MAX

Pour projeter le maillage $MA1$ sur le maillage $MA2$, la méthode cherche dans quel élément du maillage $MA1$ se trouve chaque nœud de $MA2$, puis interpole la valeur à l'aide des fonctions de forme de l'élément. Lorsque qu'un nœud de $MA2$ n'est dans aucun élément du maillage $MA1$, la méthode met en relation le nœud et le point (du bord) de l'élément le plus proche. Il interpole la valeur à l'aide des fonctions de forme de l'élément et cela même si le nœud est “loin” de cet élément.

Si l'on souhaite qu'un nœud qui n'est dans aucun des éléments du maillage $MA1$, ne soit pas concerné par la projection, on utilise l'opérande DISTANCE_MAX. Cet opérande permet de donner la distance maximale que l'on autorise entre le nœud et l'élément le plus proche.

Si le nœud ne répond pas au critère de proximité le champ ne sera pas projeté sur ce nœud (i.e. le nœud ne portera aucune composante).

Il n'y a pas de valeur par défaut pour DISTANCE_MAX. Ce qui veut dire que par défaut, le champ sera prolongé en dehors de la matière aussi loin qu'il le faudra.

4.13.3.4 DISTANCE_ALARME = d_ala

Le code émet une alarme lorsqu'un nœud de $MA2$ est jugé «éloigné » des mailles de $MA1$.
Soit d la distance séparant un nœud de $MA2$ de la maille de $MA1$ la plus proche.
Par défaut, le critère pour juger si un nœud est éloigné est relatif :
d > 1/10 ème de la taille de la maille la plus proche.
Mais si l'utilisateur utilise DISTANCE_ALARME = d_ala , le message d'alarme est émis si d > d_ala (critère absolu).

4.13.3.5 CENTRE / ANGL_NAUT / TRAN

Ces opérandes permettent de définir la transformation géométrique (rotation et/ou translation) permettant de passer de la face esclave à la face maître. La commande effectue d'abord la rotation puis la translation.

Attention : la transformation est dans le sens esclave-maître.

Cette condition aux limites s'applique aux modélisations planes ('PLAN' ou 'AXIS') ou volumiques ('3D').

4.14 Mot-clé ECHANGE_PAROI

4.14.1 But

Mot-clé facteur utilisable pour appliquer des conditions d'échange thermique entre 2 parois. Ces parois peuvent être définies de deux manières :

- chacune séparément, par une ou plusieurs mailles (cas de parois maillées) ;
- à partir d'une liste de fissures X-FEM, les deux parois correspondant alors aux deux lèvres de chaque fissure. Dans ce cas il est également possible d'imposer la continuité du champ de température à travers les lèvres de chaque fissure.

4.14.2 Syntaxe

- pour AFFE_CHAR_THER

```
ECHANGE_PAROI=_F (
```


Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

```
                        # si la paroi est maillée
                        ♦  GROUP_MA_1 =        lgma,     [l_gr_maille]
                        ♦  GROUP_MA_2 =        lgma,     [l_gr_maille]
                        ♦     COEF_H =           h,        [R]
                        ◊     TRAN =             lr,       [l_R]

                        # si la paroi est définie avec des fissures X-FEM
                        ♦     FISSURE =          lfiss,    [l_fiss_xfem]
                        ♦  /  COEF_H =           h,        [R]
                           /  TEMP_CONTINUE =    'OUI'
                    )
```

- pour AFFE_CHAR_THER_F

```
ECHANGE_PAROI=_F (
                        # si la paroi est maillée
                        ♦  GROUP_MA_1 =        lgma,     [l_gr_maille]
                        ♦  GROUP_MA_2 =        lgma,     [l_gr_maille]
                        ♦     COEF_H =           h,        [fonction]
                        ◊     TRAN =             lr,       [l_R]

                        # si la paroi est définie avec des fissures X-FEM
                        ♦     FISSURE =          lfiss,    [l_fiss_xfem]
                        ♦  /  COEF_H =           h,        [fonction]
                           /  TEMP_CONTINUE =    'OUI'
                    )
```


Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

4.14.3 Opérandes

4.14.3.1 Cas de parois maillées

♦ COEF_H

Coefficient d'échange entre les 2 parois.
Réel pour l'opérateur AFFE_CHAR_THER, fonction pour l'opérateur AFFE_CHAR_THER_F.

♦ GROUP_MA_1

♦ GROUP_MA_2

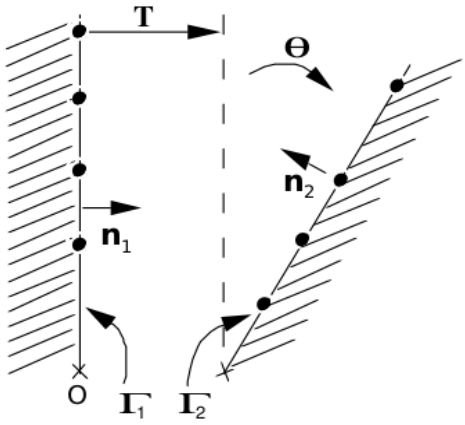


Figure 4.13.3-a

Ces opérandes permettent de définir les 2 listes de mailles représentant pour la liste indicée « _1 » la paroi Γ_1 pour la liste indicée « _2 » la paroi Γ_2.

Les parois sont en correspondance et doivent comporter le même nombre de mailles et de nœuds.

La condition limite appliquée entre ces 2 parois est :

sur Γ_1 : $k\frac{\partial T_1}{\partial n_1}=h\left(T_2-T_1\right)$ n_1 normale extérieure à Γ_1

sur Γ_2 : $k\frac{\partial T_2}{\partial n_2}=h\left(T_1-T_2\right)$ n_2 normale extérieure à Γ_2

où $T_1|_{\Gamma_1}=T \qquad T_2|_{\Gamma_1}=T\circ f$
$T_1|_{\Gamma_2}=T\circ f^{-1} \qquad T_2|_{\Gamma_2}=T$

f représentant la bijection qui met en vis-à-vis un nœud de Γ_1 et un nœud de Γ_2.

◊ TRAN

Composantes du vecteur translation.
Cette opérande permet de définir une transformation virtuelle (de translation) approximative de Γ_1 en Γ_2 afin d'assurer la bijectivité de la fonction en vis-à-vis. TRAN caractérise une translation :

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

en 2D on a donc : TRAN = (tx, ty)
en 3D on a : TRAN = (tx, ty, tz)

4.14.3.2 Cas d'une ou plusieurs fissures X-FEM

♦ FISSURE

Liste de fissures X-FEM.
Pour chaque fissure donnée dans cette liste, les deux parois en vis à vis sont définies comme la lèvre supérieure et la lèvre inférieure de la fissure. Il y a donc autant de couple de parois en vis-à-vis que de fissures présentes dans cette liste.

♦ / COEF_H

Coefficient d'échange entre les lèvres de la fissure. Réel pour l'opérateur AFFE_CHAR_THER , fonction pour l'opérateur AFFE_CHAR_THER_F .
Si ce mot-clé est présent, la condition d'échange imposée entre les lèvres de chaque fissure donnée sous le mot-clé FISSURE est la même que dans le cas d'une paroi maillée (voir § précédent), en prenant l'identitié pour la bijection f puisque les lèvres supérieure et inférieure de chaque fissure sont géométriquement confondues.

/ TEMP_CONTINUE

Ce mot-clé ne peut être renseigné qu'avec le valeur 'OUI'.
Si ce mot-clé est présent, on impose la continuité du champ de température à travers les lèvres de chaque fissure donnée sous le mot-clé FISSURE, en annulant les degrés de liberté d'enrichissement (“Heaviside” et “crack-tip”).

4.14.4 Utilisation de ECHANGE_PAROI

Dans le cas d'une paroi maillée :

L'utilisateur donne deux listes de mailles dont seront issus les couples de nœuds appariés. Ces listes sont d'abord triées par type de maille : les nœuds appariés proviendront de mailles de type identique. Pour chaque maille de la première liste, on détermine la maille la plus proche dans la deuxième liste en calculant toutes les distances des nœuds pris deux à deux (on parcourt toutes les permutations possibles). La distance minimum obtenue définit à la fois la maille en vis-à-vis et les couples de nœuds appariés pour les deux mailles concernées. Comme dans LIAISON_GROUP [§4.11], il est possible d'effectuer une transformation géométrique virtuelle (rotation et/ou translation) avant de calculer les distances.

Dans le cas d'une ou plusieurs fissures X-FEM :

On définit autant de couples de parois en vis à vis qu'il y a de fissures dans la liste renseignée pour le mot-clé FISSURE.

- Si le mot-clé COEF_H est présent, toutes les fissures contenues dans cette liste se verront affectées le même coefficient d'échange (réel ou fonction). Pour affecter une valeur du coefficient d'échange propre à chaque fissure, il faut utiliser le caractère répétable du mot-clé facteur ECHANGE_PAROI.
- Si le mot-clé TEMP_CONTINUE est présent, on impose la continuité du champ de température à travers toutes les fissures contenues dans cette liste.

4.14.1 Mailles et modélisations supportant ce chargement :

Dans le cas de parois maillées :

Les listes de mailles données par l'utilisateur doivent être constituées de mailles de bord, des mailles de couplage sont alors automatiquement générées. Le tableau ci-dessous fournit un récapitulatif des mailles de bords ainsi que des modélisations pour lesquelles ce type de chargement est supporté.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

Maille de bord	Modélisation	Maille de couplage générée
SEG2, SEG3	PLAN, PLAN_DIAG AXIS, AXIS_DIAG	SEG22, SEG33
TRIA3, TRIA6, QUAD4, QUAD8, QUAD9	3D, 3D_DIAG	TRIA33, TRIA66, QUAD44, QUAD88, QUAD99

Dans le cas d'une ou plusieurs fissures X-FEM :

Les fissures doivent préalablement être créées avec l'opérateur DEFI_FISS_XFEM [U4.82.08], et le modèle (renseigné pour le mot-clé MODELE) avec l'opérateur MODI_MODELE_XFEM [U4.41.11], ce qui n'est possible que dans les cas suivants :

- modélisation 3D pour les mailles support HEXA8, PENTA6, PYRA5, TETRA4
- modélisations PLAN et AXI pour les mailles support QUAD4 et TRIA3

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

4.15 Mot-clé LIAISON_UNIF

4.15.1 But

Mot-clé facteur permettant d'imposer une même valeur (inconnue) aux températures d'un ensemble de nœuds.

Ces nœuds sont définis par les groupes de mailles ou les groupes de nœuds auxquels ils appartiennent.

4.15.2 Syntaxe

- pour AFFE_CHAR_THER et AFFE_CHAR_THER_F

```
LIAISON_UNIF =_F (
                     ♦  /  GROUP_MA  =  lgma,            [l_gr_maille]
                        /  GROUP_NO  =  lgno,            [l_gr_noeud]

                      ♦  DDL = |  'TEMP' ,               [DEFAUT]
                               |  'TEMP_MIL',
                               |  'TEMP_INF',
                               |  'TEMP_SUP',
                   )
```

4.15.3 Opérandes

♦ / GROUP_MA
/ GROUP_NO

Ces opérandes permettent de définir une liste de n nœuds N_i dont on a éliminé les redondances (pour GROUP_MA, il s'agit des connectivités des mailles).

♦ DDL

Cet opérande permet de définir une liste de degrés de liberté $T_i(i=1,r)$ de r textes pris parmi : 'TEMP', 'TEMP_MIL', 'TEMP_INF', 'TEMP_SUP'.

Les $r\times(n-1)$ conditions 'cinématiques' résultantes sont :

$T_i(N_1)=T_i(N_k)$ pour $k\in(2,...,n)$, $i\in(1,...,r)$

Remarque :

Les composantes 'TEMP_MIL' , 'TEMP_SUP' , 'TEMP_INF' ne peuvent intervenir que pour des nœuds d'éléments de coque.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

4.16 Mot-clé LIAISON_CHAMNO

4.16.1 But

Mot-clé facteur utilisable pour définir une relation linéaire entre toutes les températures présentes dans un concept CHAM_NO.

4.16.2 Syntaxe

```
LIAISON_CHAMNO = _F (  ♦  CHAM_NO   =  chamno,            [cham_no]
                       ♦  COEF_IMPO =   b,                [R]
                       ◊  NUME_LAGR =  /   'NORMAL',      [DEFAUT]
                                       /   'APRES' ,
                    )
```

4.16.3 Opérandes

CHAM_NO =

Nom du chamno qui sert à définir la relation linéaire. Les températures reliées sont toutes celles présentes dans le chamno. Les coefficients à appliquer aux températures sont les valeurs de ces températures dans le chamno.

Exemple :

Supposons que l'on ait un chamno portant sur 3 nœuds de noms $N01$, $N02$ et $N03$.

Supposons que les valeurs des températures en ces 3 nœuds dans le chamno soient respectivement 2., 5.4 et 9.1. La relation linéaire que l'on va imposer est :
$2.T(NO1)+5.4T(NO2)+9.1T(NO3)=\beta$

COEF_IMPO =

C'est la valeur du coefficient réel β au second membre de la relation linéaire.

NUME_LAGR =

Si 'NORMAL', les 2 multiplicateurs de Lagrange associés à la relation seront tels que le premier sera situé avant tous les termes impliqués dans la relation et le second après, dans la matrice assemblée.

Si 'APRES', les 2 multiplicateurs de Lagrange associés à la relation seront situé après tous les termes impliqués dans la relation, dans la matrice assemblée.

Ce choix présente l'avantage d'avoir une matrice assemblée dont l'encombrement est plus faible mais a le désavantage de pouvoir faire apparaître une singularité dans la matrice.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

4.17 Mot-clé CONVECTION

4.17.1 But

Mot-clé utilisable pour prendre en compte le terme de transport de chaleur par convection dont l'expression est $\rho C_p . V\ grad\ T$, apparaissant dans l'expression de la dérivée particulaire $\rho C_p \frac{dT}{dt} : \rho C_p \frac{dT}{dt} = \rho C_p \frac{\partial T}{\partial t} + \rho C_p V\ grad\ T$.

Dans le cas d'un milieu liquide, V désigne la vitesse imposée de la particule fluide au point courant.

Dans le cas d'un milieu solide mobile, V désigne la vitesse du solide. Dans tous les cas, on suppose que le champ de vitesse est connu a priori. Le cas d'un solide mobile est assez fréquent en pratique. Il concerne en particulier les applications de soudage ou de traitement de surface qui mettent en jeu une source de chaleur se déplaçant dans une direction et à une vitesse données.

Le problème thermique est alors étudié dans un référentiel lié à la source (cf. THER_NON_LINE_MO [U4.54.03]).

4.17.2 Syntaxe

```
CONVECTION = _F  (  ♦  VITESSE   =  v                [cham_no_sdaster])
```

4.17.3 Opérande

Pour AFFE_CHAR_THER et AFFE_CHAR_THER_F,

VITESSE =

Nom du champ de vitesse à l'instant où l'on réalise le calcul.

Ce champ est un concept cham_no de type cham_no_depl_r. Il doit avoir été défini sur tout le modèle pour lequel on réalise le calcul.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

4.18 Mot-clé SOUR_NL

4.18.1 But

Mot-clé facteur utilisable pour appliquer des **sources volumiques dépendant de la température** (2D ou 3D) à un **domaine** défini par une ou plusieurs mailles de type **volumique**.
Ce type de flux n'est disponible que dans la commande AFFE_CHAR_THER_F et n'est utilisé que par les commandes THER_NON_LINE [U4.54.02] et THER_NON_LINE_MO [U4.54.03].

Les valeurs sont fournies par un concept de type fonction. La fonction dépend de la température, à l'exclusion de tout autre paramètre. En outre, il s'agit nécessairement d'une fonction tabulée et non d'une formule.

4.18.2 Syntaxe

```
SOUR_NL=_F    (
              ♦  /  TOUT      =          'OUI',
                 /  GROUP_MA  =          lgma,          [l_gr_maille]
              ♦  SOUR         =          sf,            [fonction]
            )
```

4.18.3 Opérandes

Ce chargement s'applique aux types de mailles et aux modélisations suivantes :

Maille	Modélisation
HEX A8, HEXA20, HEXA27 PYRA5, PYRA13, PENTA6, PENTA15 TETRA4, TETRA10	3D
TRIA3, TRIA6, QUAD4, QUAD8, QUAD9	PLAN, AXIS
HEX A8, PYRA5, PENTA6, TETRA4	3D_DIAG
TRIA3, QUAD4	PLAN_DIAG, AXIS_DIAG

/ ♦ SOUR = s,

Valeur de la source dépendant de la température et supposée constante sur l'élément.

4.19 Mot-clé EVOL_CHAR

EVOL_CHAR = evch

Le mot-clé facteur EVOL_CHAR est utilisable pour appliquer des chargements évolutifs dans le temps de type evol_char produits par LIRE_RESU [U7.02.01] ou CREA_RESU [U4.44.12] et contenant des champs de coefficients correspondant soit :

- à un chargement de type ECHANGE (voir § 4.8) : le champ T_EXT correspond au paramètre TEMP_EXT et le champ COEF_H correspond au paramètre COEF_H .
- à un chargement de type FLUX_REP (voir § 4.5) : le champ FLUN correspond au paramètre du même nom.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)