

Operators AFFE_CHAR_THER and AFFE_CHAR_THER_F

1 Goal

To affect loadings and boundary conditions thermal on a model.

For the operator `AFFE_CHAR_THER`, the affected values do not depend on any parameter and are defined by actual values.

For the operator `AFFE_CHAR_THER_F`, the values are function of one or two parameters to be chosen as a whole (`INST`, `X`, `Y`, `Z`) or of the temperature `TEMP` in nonlinear thermics.

These functions must be defined beforehand by the call to one of the operators:

- `DEFI_CONSTANTE` [U4.31.01]
- `DEFI_FONCTION` [U4.31.02]
- `DEFI_NAPPE` [U4.31.03]
- `CALC_FONC_INTERP` [U4.32.01]

The produced concept is of type `char_ther`.

Contents

1	Goal.....	1
2	General syntax.....	5
3	General information.....	6
4	Operands.....	7
4.1	General information on the operands.....	7
4.1.1	Two forms of operands under a keyword factor.....	7
4.1.2	Topological entities of assignment of the loadings.....	7
4.2	Operand MODEL.....	7
4.3	Keyword TEMP_IMPO.....	8
4.3.1	Goal.....	8
4.3.2	Syntax.....	8
4.3.3	Operands.....	8
4.4	Keyword FLUX_REP.....	10
4.4.1	Goal.....	10
4.4.2	Syntax.....	10
4.4.3	Operands.....	10
4.4.4	Notice.....	11
4.5	Keyword FLUX_NL.....	13
4.5.1	Goal.....	13
4.5.2	Syntax.....	13
4.5.3	Operands.....	13
4.6	Keyword RADIATION.....	14
4.6.1	Goal.....	14
4.6.2	Syntax.....	14
4.6.3	Operands.....	14
4.7	Keyword EXCHANGE.....	15
4.7.1	Goal.....	15
4.7.2	Syntax.....	15
4.7.3	Operands.....	16
4.8	Keyword SOURCE.....	17
4.8.1	Goal.....	17
4.8.2	Syntax.....	17
4.8.3	Operands.....	17
4.9	Keyword PRE_GRAD_TEMP.....	18
4.9.1	Goal.....	18
4.9.2	Syntax.....	18
4.9.3	Operands.....	18
4.10	Keyword LIAISON_DDL.....	20

4.10.1	Goal.....	20
4.10.2	Syntax.....	20
4.10.3	Operands.....	20
4.11	Keyword LIAISON_GROUP.....	23
4.11.1	Goal.....	23
4.11.2	Syntax.....	23
4.11.3	Operands.....	24
4.11.4	Use of LIAISON_GROUP.....	27
4.12	Keyword LIAISON_MAIL.....	28
4.12.1	Goal.....	28
4.12.2	Syntax.....	28
4.12.3	Operands.....	28
4.12.3.1	GROUP_MA_ESCL / MAILLE_ESCL / GROUP_NO_ESCL / NOEUD_ESCL.....	28
4.12.3.2	GROUP_MA_MAIT / MAILLE_MAIT.....	28
4.12.3.3	DISTANCE_MAX.....	29
4.12.3.4	CENTER / ANGL_NAUT / TRAN.....	29
4.13	Keyword ECHANGE_PAROI.....	29
4.13.1	Goal.....	29
4.13.2	Syntax.....	29
4.13.3	Operands.....	31
4.13.3.1	Case of walls with a grid.....	31
4.13.3.2	Case of one or more cracks X-FEM.....	32
4.13.4	Use of ECHANGE_PAROI.....	32
4.13.5	Meshs and modelings supporting this loading:.....	32
4.14	Keyword LIAISON_UNIF.....	34
4.14.1	Goal.....	34
4.14.2	Syntax.....	34
4.14.3	Operands.....	34
4.15	Keyword LIAISON_CHAMNO.....	35
4.15.1	Goal.....	35
4.15.2	Syntax.....	35
4.15.3	Operands.....	35
4.16	Keyword CONVECTION.....	36
4.16.1	Goal.....	36
4.16.2	Syntax.....	36
4.16.3	Operand.....	36
4.17	Keyword SOUR_NL.....	37
4.17.1	Goal.....	37
4.17.2	Syntax.....	37
4.17.3	Operands.....	37

4.18Keyword EVOL_CHAR.....37

2 General syntax

```
CH [char_ther] = AFFE_CHAR_THER

( ♦ MODEL = Mo, [model]

  ♦ | TEMP_IMPO = (see keyword TEMP_IMPO )
    | FLUX_REP = (see keyword FLUX_REP )
    | RADIATION = (see keyword RADIATION )
    | EXCHANGE = (see keyword EXCHANGES )
    | SOURCE = (see keyword SOURCE )
    | PRE_GRAD_TEMP = (see keyword PRE_GRAD_TEMP )
    | LIAISON_DDL = (see keyword LIAISON_DDL )
    | LIAISON_GROUP = (see keyword LIAISON_GROUP )
    | LIAISON_MAIL = (see keyword LIAISON_MAIL )
    | ECHANGE_PAROI = (see keyword ECHANGE_PAROI )
    | LIAISON_UNIF = (see keyword LIAISON_UNIF )
    | LIAISON_CHAMNO = (see keyword LIAISON_CHAMNO )
    | CONVECTION = (see keyword CONVECTION )
    | EVOL_CHAR = (see keyword EVOL_CHAR )
  )
```

```
CH [char_ther] = AFFE_CHAR_THER_F

( ♦ MODEL = Mo, [model]

  ♦ | TEMP_IMPO = (see keyword TEMP_IMPO )
    | FLUX_REP = (see keyword FLUX_REP )
    | FLUX_NL = (see keyword FLUX_NL )
    | RADIATION = (see keyword RADIATION )
    | EXCHANGE = (see keyword EXCHANGES )
    | SOURCE = (see keyword SOURCE )
    | PRE_GRAD_TEMP = (see keyword PRE_GRAD_TEMP )
    | LIAISON_DDL = (see keyword LIAISON_DDL )
    | LIAISON_GROUP = (see keyword LIAISON_GROUP )
    | ECHANGE_PAROI = (see keyword ECHANGE_PAROI )
    | LIAISON_UNIF = (see keyword LIAISON_UNIF )
    | CONVECTION = (see keyword CONVECTION )
    | SOUR_NL = (see keyword SOUR_NL )
  )
```

3 General information

Possible error messages related to the order AFFE_CHAR_THER

It happens sometimes that a thermal ordering of calculation (THER_LINEAIRE, THER_NON_LINE,...) stop in fatal error during the calculation of the second elementary members due to the loadings defined in the orders AFFE_CHAR_THER_xx.

When the code stops during these elementary calculations, important information of the error message is the name of the option of calculation required by the code. The name of this option is in general unknown to the user and it is thus difficult for him to understand the message.

In the table below, one gives in with respect to the names of the options of calculation, the name of the order and the keyword factor which make it possible to activate this option.

Elementary option of calculation	Order	Keyword factor
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	EXCHANGE
CHAR_THER_TEXT_R	AFFE_CHAR_THER	EXCHANGE
CHAR_THER_SOURNL	AFFE_CHAR_THER_F	SOUR_NL

4 Operands

4.1 General information on the operands

4.1.1 Two forms of operands under a keyword factor

The operands under a keyword factor are of two forms:

- operands specifying the topological entities where the loadings (keywords are affected `GROUP_NO` and `GROUP_MA`, etc...). The arguments of these operands are identical for the two operators.
- operands specifying the affected values (`TEMP`, `COEF_H`, etc...). The significance of these operands is the same one for the two operators but the arguments of these operands are all of the real type for the operator `AFFE_CHAR_THER` and of the type `function` (created by one of the operators `DEFI_FONCTION`, `DEFI_NAPPE`, `DEFI_CONSTANTE` or `CALC_FONC_INTERP`) for the operator `AFFE_CHAR_THER_F`.

We will thus not distinguish in this document, except fast mention of the opposite, the two operators `AFFE_CHAR_THER` and `AFFE_CHAR_THER_F`.

4.1.2 Topological entities of assignment of the loadings

In a general way, the topological entities on which values must be affected are defined:

- by nodes and in this case:
 - maybe by the operand `GROUP_NO` allowing to introduce a list of group of nodes,
 - maybe by the operand `NODE` allowing to introduce a list of nodes.
- by mesh and in this case:
 - maybe by `GROUP_MA` allowing to introduce a list of groups of meshes,
 - maybe by `MESH` allowing to introduce a list of meshes.

Rule:

To define the field of assignment most simply possible, one uses the rule of overload it is the last assignment which precedes.

4.2 Operand MODEL

◆ `MODEL = Mo,`

Concept produced by the operator `AFFE_MODELE` [U4.41.01] where are defined the types of finite elements affected on the grid.

4.3 Keyword TEMP_IMPO

4.3.1 Goal

Keyword factor usable to impose, on nodes or groups of nodes, a temperature.

According to the name of the operator called, the values are provided directly (AFFE_CHAR_THER) or via a concept of the type function (AFFE_CHAR_THER_F).

4.3.2 Syntax

- for AFFE_CHAR_THER

```
TEMP_IMPO = _F ( ♦ | ALL      = 'YES',
                  | NODE      = lno,           [l_noeud]
                  | GROUP_NO = lgno,          [l_gr_noeud]
                  | MESH     = lma,           [l_maille]
                  | GROUP_MA = lgma,          [l_gr_maille]
                  ♦ / TEMP      = T,           [R]
                  / | TEMP_MIL = tinf,         [R]
                  | TEMP_INF  = tinf,         [R]
                  | TEMP_SUP  = tsup,         [R]
                  )
```

- for AFFE_CHAR_THER_F

```
TEMP_IMPO = _F ( ♦ | ALL      = 'YES',
                  | NODE      = lno,           [l_noeud]
                  | GROUP_NO = lgno,          [l_gr_noeud]
                  | MESH     = lma,           [l_maille]
                  | GROUP_MA = lgma,          [l_gr_maille]
                  ♦ / TEMP      = tf,          [function]
                  / | TEMP_MIL = tf,          [function]
                  | TEMP_INF  = tinf,         [function]
                  | TEMP_SUP  = tsupf,        [function]
                  )
```

4.3.3 Operands

/ TEMP =

Value of **temperature** imposed on (S) the node (S) specified (S).

/ For the elements of thermal hull only (Modeling: 'HULL') :

| TEMP_MIL

Temperature on the average layer imposed on (S) the node (S) specified (S).

| TEMP_INF

Temperature imposed on the lower wall of the hull.

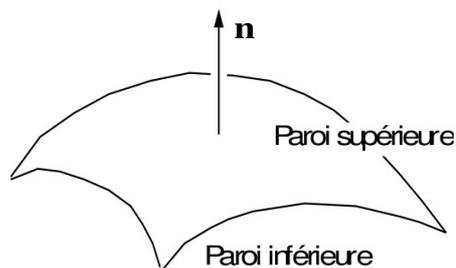
| TEMP_SUP

Temperature imposed on the higher wall of the hull.

These options make it possible to represent a parabolic variation of the temperature in the thickness.

Note:

*The hull is directed by the connectivity of the nodes of the associated mesh (cf [U3.01.00]).
That is to say n the normal vector directing the hull:*



4.4 Keyword FLUX_REP

4.4.1 Goal

Keyword factor usable to apply **normal flows**, with one **face** of voluminal element or thermal hull defined by one or more meshes or of the groups of meshes of the type **triangle** or **quadrangle**. This keyword also makes it possible to apply a normal flow to an edge (in 2D PLAN or AXIS or AXIS_FOURIER) on meshes of type segment.

According to the name of the operator called, the values are provided directly (AFFE_CHAR_THER) or via a concept of the type function (AFFE_CHAR_THER_F).

4.4.2 Syntax

- for AFFE_CHAR_THER

```

FLUX_REP = _F (
    ◆ / ALL      =      'YES',
    ◇ | MESH     =      lma,          [l_maille]
      | GROUP_MA =      lgma,        [l_gr_maille]

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

- for AFFE_CHAR_THER_F

```

FLUX_REP = _F (
    ◆ / ALL      =      'YES',
    ◇ | MESH     =      lma,          [l_maille]
      | GROUP_MA =      lgma,        [l_gr_maille]

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

4.4.3 Operands

/ FLUN: fl normal flow imposed on the mesh.

This loading applies to the types of meshes and following modelings:

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

More precisely the boundary condition applied is: $\lambda(\text{grad } T \cdot \mathbf{n}) = f_l$

where λ is thermal conductivity and \mathbf{n} is the normal directed in the direction of the classification of the nodes of the mesh. The convention of orientation is that used in AFFE_CHAR_MECA [U4.44.01].

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

Normal flow imposed on the walls lower and higher of a thermal hull.

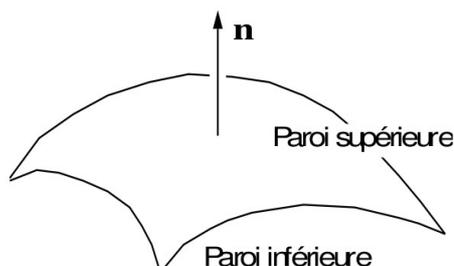
These loadings apply to the types of meshes and following modelings:

Mesh	Modeling
TRIA3, TRIA6	HULL

\mathbf{n} being the normal directing surface [U4.44.01], the boundary condition applied is:

$\lambda(\text{grad } T \cdot \mathbf{n}) = f_{lin}$ where f_{lin} is the normal flow imposed on the lower wall of the hull,

$\lambda(\text{grad } T \cdot \mathbf{n}) = f_{lsup}$ where f_{lsup} is the normal flow imposed on the higher wall of the hull.



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

Vectorial flow \mathbf{fl} in the total reference mark (only for AFFE_CHAR_THER_F) that one projects on the normal with the element (for the definition of the normal [U4.44.01]).

$$\lambda(\text{grad } T \cdot \mathbf{n}) = \mathbf{fl} \cdot \mathbf{n} = fl_x.n_x + fl_y.n_y + fl_z.n_z$$

This loading applies to the types of meshes and modelings:

Mesh	Modeling
SEG2, SEG3	PLAN PLAN_DIAG

Note: the rule of remanence (see U1.03.00) applies between the various quantities which one can affect: FLUN, FLUN_INF, ... FLUX_Z.

4.4.4 Notice

The simple keyword CARA_TORSION this keyword factor FLUX_REP is not documented here and does not have to be employed by the user. He is used only for the macro-order MACR_CARA_POUTRE. This one is used to identify the geometrical characteristics of the sections of beams. For the characteristics

of torsion, the order solves a problem of Laplacian by employing in an indirect way the operators of linear thermics.

4.5 Keyword FLUX_NL

4.5.1 Goal

Keyword factor usable to apply **normal flows** functions of the temperature, with one **face** of voluminal element defined by one or more meshes or of the groups of meshes of the type **triangle** or **quadrangle**. This keyword also makes it possible to apply a normal flow to an edge (in 2D PLAN or AXIS) on meshes of type segment. One can thus model a condition of radiation of the standard law of STEPHAN. This kind of flow is used only by the orders THER_NON_LINE [U4.54.02] and THER_NON_LINE_MO [U4.54.03].

The values are provided by a concept of the type function.

4.5.2 Syntax

- For AFFE_CHAR_THER_F

```

FLUX_NL = _F (
    ♦ / ALL      =      'YES',
      | MESH     =      lma,           [l_maille]
      | GROUP_MA =      lgma,         [l_gr_maille]
    ♦ FLUN      =      fl,           [function]
)

```

4.5.3 Operands

FLUN: normal flow imposed on the mesh.

This loading applies to the types of meshes and following modelings:

Mesh	Modeling
TRIA3, TRIA6, QUAD4, QUAD8, QUAD9	3D, 3D_DIAG
SEG2, SEG3	PLAN, AXIS PLAN_DIAG, AXIS_DIAG

More precisely the boundary condition applied is:

$$\lambda(\text{grad } T \cdot \mathbf{n}) = f_l$$

where \mathbf{n} is the normal directed in the direction of the classification of the nodes of the mesh. Orientation used in AFFE_CHAR_MECA document [U4.44.01].

4.6 Keyword RADIATION

4.6.1 Goal

Keyword allowing to define the flow radiated ad infinitum according to the formula:

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

by the data of emissivity ϵ , the Boltzmann constant σ and the temperature ad infinitum T_{∞} expressed into Centigrade. The temperature T will be also expressed into Centigrade, it is thus necessary to take care, by coherence, to use only degrees Celsius for all the study.

4.6.2 Syntax

- for AFFE_CHAR_THER

```

RADIATION = _F (
    ◆ / ALL      =      'YES',
      ◇ | MESH    =      lma,          [l_maille]
        | GROUP_MA =      lgma,       [l_gr_maille]

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

```
- for AFFE_CHAR_THER_F

```

RADIATION = _F (
    ◆ / ALL      =      'YES',
      ◇ | MESH    =      lma,          [l_maille]
        | GROUP_MA =      lgma,       [l_gr_maille]

    ◆ SIGMA     =      sigma,         [function]
    ◆ EPSILON   =      epsilon,       [function]
    ◆ TEMP_EXT=      tex,             [function]
)

```

4.6.3 Operands

- ◆ SIGMA = sigma
- ◆ EPSILON = epsilon
- ◆ TEMP_EXT = tex

This loading applies to following modelings:

Mesh	Modeling
TRIA3, TRIA6, QUAD4, QUAD8, QUAD9	3D, 3D_DIAG
SEG2, SEG3	PLAN, AXIS PLAN_DIAG, AXIS_DIAG

sigma: Boltzmann constant, $\sigma = 5.6710^{-8}$ in units IF ($W/m^2.K^4$) (attention with this value if the units of grid change),

epsilon: emissivity,

tex: temperature ad infinitum in degrees Celsius.

4.7 Keyword EXCHANGE

4.7.1 Goal

Keyword factor usable to apply **conditions of exchange** with an outside temperature with one **face** voluminal elements or hulls, defined by one or more meshes or of the groups of meshes of the type **triangle** or **quadrangle**. This keyword also makes it possible to apply conditions of exchange to an edge (in 2D PLAN or AXIS or AXIS_FOURIER) on meshes of type segment.

According to the name of the operator called, the values are provided directly (AFFE_CHAR_THER) or via a concept of the type function (AFFE_CHAR_THER_F).

4.7.2 Syntax

- for AFFE_CHAR_THER

```

EXCHANGE = _F (
    ♦ / ALL      =      'YES' ,
      / | MESH    =      lma,          [l_maille]
        | 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]
      )

```

- for AFFE_CHAR_THER_F

```

EXCHANGE = _F (
    ♦ / ALL      =      'YES' ,
      / | MESH    =      lma,          [l_maille]
        | GROUP_MA =      lgma,       [l_gr_maille]

    ♦ / ♦ COEF_H  =      HF,          [function]
      ♦ TEMP_EXT =      texf,         [function]

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

```

4.7.3 Operands

- / ♦ COEF_H = H,
- ♦ TEMP_EXT = tex,

This loading applies to the types of meshes and following modelings:

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

More precisely the boundary condition applied is:

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

where \mathbf{n} is the normal directed in the direction of the classification of the nodes tops (orientation used in AFFE_CHAR_MECA [U4.44.01]).

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

This loading applies to the types of meshes and following modelings:

Mesh	Modeling
TRIA3, TRIA6	HULL

\mathbf{n} being the normal directing surface [U2.03.03], the boundary condition applied is:

$$\lambda(\text{grad } T \cdot \mathbf{n}) = \text{hin}(\text{texin} - T)$$

where hin coefficient of exchange on the lower wall of the hull,
 and texin outside temperature, with dimensions lower wall.

$$\lambda(\text{grad } T \cdot \mathbf{n}) = \text{hsup}(\text{texsup} - T)$$

where hsup coefficient of exchange on the higher wall of the hull,
 and outside temperature, with dimensions external wall.
 texsup

Note:

- the rule of remanence (see U1.03.00) applies between the various quantities which one can affect: COEF_H, COEF_H_INF, ... TEMP_EXT_SUP.
- In the case as of loadings functions (AFFE_CHAR_THER_F), as there are two distinct components (the coefficient of exchange and the temperature), which are a priori independent, one cannot simultaneously use this loading with a multiplying function in the operator of resolution (THER_LINEAIRE or THER_NON_LINE). An error message will inform you.

4.8 Keyword SOURCE

4.8.1 Goal

Keyword factor usable to apply **voluminal sources** (2D or 3D) with one **field** defined by one or more meshes or of the groups of meshes of the type **voluminal**.

According to the name of the operator called, the values are provided directly (AFFE_CHAR_THER) or via a concept of the type *function* (AFFE_CHAR_THER_F).

4.8.2 Syntax

- for AFFE_CHAR_THER

```
SOURCE=_F (
    ♦ / ALL = 'YES',
      / | MESH = lma, [l_maille]
        | GROUP_MA = lgma, [l_gr_maille]
    ♦ ..... / SOUR = S, [R]
      / SOUR_CALCULEE = chs, [cham_elem]
)
```

- for AFFE_CHAR_THER_F

```
SOURCE=_F (
    ♦ / ALL = 'YES',
      / | MESH = lma, [l_maille]
        | GROUP_MA = lgma, [l_gr_maille]
    ♦ SOUR = sf, [function]
)
```

4.8.3 Operands

This loading applies to the types of meshes and following modelings:

Mesh	Modeling
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,

Value of the presumedly constant source on the element.

/ ♦ SOUR_CALCULEE = chs,

Name of *cham_elem_sour_R* containing on each element the values of the source discretized at the points of Gauss (1^{era} family).

4.9 Keyword PRE_GRAD_TEMP

4.9.1 Goal

Keyword factor usable to apply to an element 3D or 2D (PLAN, AXIS) a presumed uniform variation in temperature in the element. This "initial" variation in temperature is usable for example to solve the elementary problems determining the correctors of stationary linear thermics in the basic cell (2D, 3D), in periodic homogenisation.

The coefficients of homogenized conductivity are obtained while calculating by the operator POST_ELEM [U4.81.22] keyword ENER_POT the energy dissipated thermically with balance in linear thermics starting from the correctors.

Because of the thermal analogy, this approach can be exploited to obtain the correctors in elasticity antiplane in the basic cell 2D, as well as in electric conduction.

Calculation uses the conductivity of material. This one is supposed to be isotropic and independent of the temperature.

The assignment can be done on one or more meshes, one or more groups of meshes or on all the elements of the model.

4.9.2 Syntax

- for AFFE_CHAR_THER

```
PRE_GRAD_TEMP = _F (
    ◆ / ALL = 'YES',
      / | MESH = lma, [l_maille]
      | GROUP_MA = lgma, [l_gr_maille]
    ◆ | FLUX_X = flx, [R]
      | FLUX_Y = fly, [R]
      | FLUX_Z = flz, [R]
    )
```

4.9.3 Operands

This loading applies to the types of meshes and following modelings:

Mesh	Modeling
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) (in 3D only)

Components of the variation in temperature $grad T_{ini}$ in the total reference mark.

The second calculated elementary member is: $\int_{V_e} grad T_{ini} K grad T^* dV_e$ where K is the tensor of thermal conductivities.

The gradients can be a function of the geometry and/or time.

- for AFFE_CHAR_THER_F
PRE_GRAD_TEMP = _F (
 - ◆ / ALL = 'YES',
 - / MESH = lma, [l_maille]
 - / GROUP_MA = lgma, [l_gr_maille]
 - ◆ | FLUX_X = flxf, [function]
 - | FLUX_Y = flyf, [function]
 - | FLUX_Z = flzf, [function])

4.10 Keyword LIAISON_DDL

4.10.1 Goal

Keyword factor usable to define a linear relation between degrees of freedom of two or several nodes.

According to the name of the operator called, the values are provided directly (AFFE_CHAR_THER) or via a concept function (AFFE_CHAR_THER_F).

4.10.2 Syntax

- for AFFE_CHAR_THER

```
LIAISON_DDL = _F ( ♦ / NODE = lno, [l_noeud]
                  / GROUP_NO = lgno, [l_gr_noeud]

                  ◇ DDL = | 'TEMP', [DEFECT]
                          | 'TEMP_MIL',
                          | 'TEMP_INF',
                          | 'TEMP_SUP',
                          | 'H1',

                  ♦ COEF_MULT = hasI , [l_R]
                  ♦ COEF_IMPO = B , [R]
                )
```

- for AFFE_CHAR_THER_F

```
LIAISON_DDL = _F ( ♦ / NODE = lno, [l_noeud]
                  / GROUP_NO = lgno, [l_gr_noeud]

                  ◇ DDL = | 'TEMP', [DEFECT]
                          | 'TEMP_MIL',
                          | 'TEMP_INF',
                          | 'TEMP_SUP',
                          | 'H1',

                  ♦ COEF_MULT = hasI , [l_R]
                  ♦ COEF_IMPO = BF , [function]
                )
```

4.10.3 Operands

The list of the nodes N_i ($i=1,r$) defined by GROUP_NO or NODE is ordered in a natural way:

- in the order of the list of group of nodes, and for each group of nodes, in the order of definition of the group by GROUP_NO.
- in the order of the list of nodes for NODE.

The argument of DDL must be a list of degrees of freedom T_i ($i=1,r$) of r texts taken among: 'TEMP', 'TEMP_MIL', 'TEMP_SUP', 'TEMP_INF', 'H1'

If the keyword DDL is omitted, by default the linear relation will relate to the degrees of freedom 'TEMP'.

The argument of COEF_MULT must be a list a_i ($i=1,r$) coefficients (of real type for AFFE_CHAR_THER and AFFE_CHAR_THER_F).

The argument of COEF_IMPO is a coefficient β for AFFE_CHAR_THER, a function of space for AFFE_CHAR_THER_F.

The following kinematic condition is applied:
$$\sum_{i=1}^r \alpha_i T_i = \beta$$

Note:

Components 'TEMP_MIL', 'TEMP_SUP' and 'TEMP_INF' can intervene only in affected combinations **only** with nodes which belong to elements of **hull** (modeling 'HULL').

The component 'H1' can intervene only in affected combinations **only** with nodes which belong to elements **X-FEM**. In this case, only degrees of freedom 'TEMP' and 'H1' can appear in the linear relation.

In the case of a linear relation between the degrees of freedom of the same node, one will repeat behind the keyword *NODE* the name of the node as many times as there are degrees of freedom in the relation. **Example:** to impose $T_{\text{sup}} = T_{\text{inf}}$ on the node *NI*, one will write:

```
LIAISON_DDL = _F (  NODE      = (N1, N1),
                   DDL       = ('TEMP_SUP', 'TEMP_INF'),
                   COEF_MULT = (1. , - 1.),
                   COEF_IMPO = 0. ,
                   )
```

It is important to note that with an occurrence of the keyword factor *LIAISON_DDL* corresponds one and only one linear relation.

If one wants to impose the same relation between 2 groups of nodes *GRN01* and *GRN02* (even temperature node with node for example) **one cannot write:**

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

This writing has direction only if *GRN01* and *GRN02* contain each one one node. It will be necessary in the case to clarify each linear relation above, node by node.

The keyword *LIAISON_GROUP* on the other hand allows to condense the writing of the linear relations between 2 groups of nodes in opposite.

4.11 Keyword LIAISON_GROUP

4.11.1 Goal

Keyword factor usable to define linear relations between couples of nodes, these couples of nodes being obtained while putting in opposite two lists of meshes or nodes.

According to the name of the operator called N , the values are provided directly (AFFE_CHAR_THER) or via a concept function (AFFE_CHAR_THER_F).

4.11.2 Syntax

- for AFFE_CHAR_THER

```

LIAISON_GROUP=_F (  ◆ / ◆ / MAILLE_1      = lma1, [1_maille]
                   /   /   GROUP_MA_1     = lgma1, [1_gr_maille]

                   ◆ / MAILLE_2      = lma2, [1_maille]
                   /   /   GROUP_MA_2     = lgma2, [1_gr_maille]

                   /   ◆ / NOEUD_1      = lno1, [1_noeud]
                   /   /   GROUP_NO_1    = lgnol, [1_gr_noeud]

                   ◆ / NOEUD_2      = lno2, [1_noeud]
                   /   /   GROUP_NO_2    = lgnol, [1_gr_noeud]

                   ◇ / SANS_NOEUD      = lno, [1_noeud]
                   /   / SANS_GROUP_NO  = lgnol, [1_gr_noeud]

                   ◇ DDL_1 = |         `TEMP',           [DEFECT]
                           |         `TEMP_MIL',
                           |         `TEMP_INF',
                           |         `TEMP_SUP',
                           |         `H1',

                   ◇ DDL_2 = |         `TEMP',           [DEFECT]
                           |         `TEMP_MIL',
                           |         `TEMP_INF',
                           |         `TEMP_SUP',
                           |         `H1',

                   ◆ COEF_MULT_1 =     has1i ,           [1_R]
                   ◆ COEF_MULT_2 =     has2i ,           [1_R]
                   ◆ COEF_IMPO      =     B ,           [R]

                   ◇ | CENTER      = Lr ,           [1_R]
                   | ANGL_NAUT     = Lr ,           [1_R]
                   | TRAN          = Lr ,           [1_R]

                   ◇ TOP          = `YES',

                   )

```

```

• for AFFE_CHAR_THER_F
    LIAISON_GROUP=_F (
        ◆ / ◆ / MAILLE_1 = lma1, [l_maille]
                / GROUP_MA_1 = lgma1, [l_gr_maille]
        ◆ / MAILLE_2 = lma2, [l_maille]
                / GROUP_MA_2 = lgma2, [l_gr_maille]
        / ◆ / NOEUD_1 = lno1, [l_noeud]
                / GROUP_NO_1 = lgno1, [l_gr_noeud]
        ◆ / NOEUD_2 = lno2, [l_noeud]
                / GROUP_NO_2 = lgno2, [l_gr_noeud]
    ◇ / SANS_NOEUD = lno, [l_noeud]
        / SANS_GROUP_NO = lgno, [l_gr_noeud]
    ◇ DDL_1 = | 'TEMP', [DEFECT]
              | 'TEMP_MIL',
              | 'TEMP_INF',
              | 'TEMP_SUP',
              | 'H1',
    ◇ DDL_2 = | 'TEMP', [DEFECT]
              | 'TEMP_MIL',
              | 'TEMP_INF',
              | 'TEMP_SUP',
              | 'H1',
    ◆ COEF_MULT_1 = has1i, [l_R]
    ◆ COEF_MULT_2 = has2i, [l_R]
    ◆ COEF_IMPO = BF, [function]
    ◇ | CENTER = Lr, [l_R]
      | ANGL_NAUT = Lr, [l_R]
      | TRAN = Lr, [l_R]
    ◇ TOP = 'YES'
    )
    
```

4.11.3 Operands

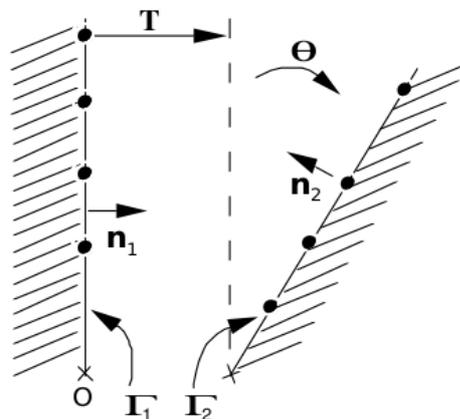


Figure 4.11.3-a: Geometrical transformation of a border in another

Kinematic condition "general":
$$\sum_{i=1}^{NDDL1} \alpha_{1i} T_i|_{\Gamma_1} + \sum_{i=1}^{NDDL2} \alpha_{2i} T_i|_{\Gamma_2} = \beta$$

/ ♦ / MAILLE_1 =
/ GROUP_MA_1 =

These operands define Γ_1 via the meshes which compose it.

♦ / MAILLE_2 =
/ GROUP_MA_2 =

These operands define Γ_2 via the meshes which compose it.

/ ♦ / NOEUD_1 =
/ GROUP_NO_1 =

These operands define Γ_1 via the nodes which compose it.

♦ / NOEUD_2 =
/ GROUP_NO_2 =

These operands define Γ_2 via the nodes which compose it.

◇ / SANS_GROUP_NO =:
/ SANS_NOEUD =

These operands make it possible to remove list of the couples of nodes out of screw - with - screw all the couples of which at least one of the nodes belongs to the list of nodes described by these operands.

That makes it possible to avoid the accumulation of linear relations on the same node during various iterations on the keyword factor LIAISON_GROUP what leads most of the time to a singular matrix.

♦ COEF_MULT_1 (resp. COEF_MULT_2)

List of realities dimensioned exactly with the number of degrees of freedom declared in DDL_1 (resp. DDL_2) corresponding to the multiplying coefficients of the linear relation.

♦ COEF_IMPO: coefficient of blocking of the linear relation:

β : reality for AFFE_CHAR_THER
 β_f : function for AFFE_CHAR_THER_F

◇ CENTER : coordinates of the centre of rotation

◇ ANGL_NAUT : nautical angles in degrees defining rotation (see AFFE_CARA_ELEM [U4.42.01] keyword ORIENTATION)

◇ TRAN : components of the vector translation

These operands make it possible to define a virtual transformation (rotation and/or translation) approximate of Γ_1 in Γ_2 in order to ensure the bijectivity of the function opposite.

◇ DDL_1 (resp. DDL_2) :

List of texts taken among:

'TEMP', 'TEMP_MIL', 'TEMP_INF', 'TEMP_SUP', 'H1'

'TEMP_MIL', 'TEMP_INF' and 'TEMP_SUP' can be used only for elements of thermal hull (modeling: 'HULL').

'H1' can be used only for elements X-FEM. In this case, only degrees of freedom 'TEMP' and 'H1' can appear in the linear relations.

By default, the degree of freedom considered for all the nodes of the linear relations is `TEMP`.

◇ TOP = 'YES'

When the meshes of edge are quadratic, the use of TOP: 'YES' force the algorithm of pairing to associate the nodes tops with other nodes tops. In the case of fine grids, that makes it possible in certain cases to avoid the problems of conflicts of screw - with-screw.

4.11.4 Use of LIAISON_GROUP

- LIAISON_GROUP generate linear relations only between 2 nodes (one on Γ_1 , one on Γ_2)

To generate linear relations on more than 2 nodes, to use the keyword LIAISON_DDL.

- determination of the couples of nodes in opposite:

initially, one draws up the two lists of nodes to be put in opposite (IE to be paired), for each occurrence of the keyword factor LIAISON_GROUP :

- for the keywords GROUP_NO_1 and GROUP_NO_2, they are the nodes setting up the groups of nodes,
- for the keywords GROUP_MA_1 and GROUP_MA_2, they are the nodes of the meshes setting up the groups of meshes.

The redundancies being eliminated, the two lists of nodes obtained must have the same length.

The determination of the couples of nodes in opposite is done in several stages:

- for each node $N1$ first list, one seeks the node image $N2 = f(N1)$ second list. If F is not injective (a node $N2$ is the image of two distinct nodes $N1$ and $N1'$), the following error message is transmitted:

```
<F> <AFFE_CHAR_THER> <PACOAP> CONFLICT IN OPPOSITE  
NODES  
THE NODE N2 IS IT WITH RESPECT TO THE NODES N1 AND N1'
```

- for each node $N2$ second list, one seeks the node image $N1 = g(N2)$ first list. If G is not injective (a node $N1$ is the image of two distinct nodes $N2$ and $N2'$), the following error message is transmitted:

```
<F> <AFFE_CHAR_MECA> <PACOAP> CONFLICT IN OPPOSITE  
NODES  
THE N1 NODE IS IT WITH RESPECT TO THE NODES N2 AND N2'
```

- it is checked that $g = f^{-1}$, i.e. the couples obtained by the stages a) and b) are the same ones (one wants to have a bijection f between the two lists of nodes). If F is not surjective, the following error message is transmitted:

```
<F> <AFFE_CHAR_MECA> <PACOAP> CONFLICT IN OPPOSITE GENERATE  
SUCCESSIVELY FROM LISTS LIST1 AND LIST2  
THE NODE OF THE FIRST N1 LIST IS NOT THE IMAGE OF ANY NODE BY  
OPPOSITE CORRESPONDENCE
```

For a node N given, node image is called $f(N)$ the node of the other list of nodes which carries out the minimum of distance with N . To facilitate pairing, in particular in the case of particular geometries (where borders Γ_1 and Γ_2 could "almost" result one from the other by the composition of a translation and of a rotation), one gives the opportunity of making a virtual geometrical transformation of the first group of nodes (translation and rotation (cf [Figure 4.11.3-a]) before calculating the distances (keywords TRAN, CENTER and ANGL_NAUT).

For each occurrence of the keyword factor LIAISON_GROUP, one thus builds the list of the new couples in opposite. When all the occurrences were swept, one removes list the couples in double.

Note:

In the couples of nodes in opposite, the order of the nodes is important. So for the first occurrence of `LIAISON_GROUP`, a node N belonged to the first group of nodes and a node M with the second group of nodes, and that for the second occurrence of `LIAISON_GROUP`, it is the reverse, one will obtain at the conclusion of pairing the couples (N, M) and (M, N) . They will not be eliminated during detection of the redundancies; on the other hand, the matrix obtained will be singular. Thus, one advises to keep same logic during the description of the edges out of screw - with - screw.

4.12 Keyword `LIAISON_MAIL`

4.12.1 Goal

Keyword factor allowing “to thermically restick” two edges of a structure. These edges can be with a grid differently (incompatible grids) but must result one from the other by rotation and/or translation.

4.12.2 Syntax

- in `AFFE_CHAR_THER` only

```
LIAISON_MAIL =_F (  ◆ |  GROUP_MA_MAIT =          lgma_mait,
                   |  MAILLE_MAIT  =          lma_mait,

                   ◆ |  GROUP_MA_ESCL =          lgma_escl,
                   |  MAILLE_ESCL  =          lma_escl,
                   |  GROUP_NO_ESCL =          lgnö_escl,
                   |  NOEUD_ESCL   =          lno_escl,

                   ◇   DISTANCE_MAX =          d_max,          [R]

                   ◇ |  ◆ TRAN   =          (tx, ty, [tz]),          [l_R]
                   |  ◆ CENTER =          (xc, yc, [zc]),          [l_R]
                   |  ◆ ANGL_NAUT =          (alpha, [beta, gamma]), [l_R]
                   )
```

Face 1 is called face “Master”; face 2 is called face “slave”.

4.12.3 Operands

4.12.3.1 `GROUP_MA_ESCL / MAILLE_ESCL / GROUP_NO_ESCL / NOEUD_ESCL`

These keywords make it possible to define the whole of the nodes of the face slave. One takes all the nodes specified by the keywords `GROUP_NO_ESCL` and `NOEUD_ESCL` more possibly nodes carried by the meshes specified by the keywords `GROUP_MA_ESCL` and `MAILLE_ESCL`.

4.12.3.2 `GROUP_MA_MAIT / MAILLE_MAIT`

These keywords make it possible to define the whole of the meshes where they with respect to the nodes of the face slave will be sought.

One should not give the meshes of surface (in 3D) composing the adjacent face Master, but voluminal meshes with the face Master. The specified meshes are candidates for the research of opposite. One can give too much of it.

4.12.3.3 DISTANCE_MAX

To project the grid *MA1* on the grid *MA2*, the method seeks in which element of the grid *MA1* each node is of *MA2*, then the value using the functions of form of the element interpolates. When that a node of *MA2* is not in any element of the grid *MA1*, the method connects the node and the point (of the edge) of the element nearest. It interpolates the value using the functions of form of the element and that even if the node is "far" from this element.

If it be wished that a node which is not in any the elements of the grid *MA1*, is not concerned with projection, one uses the operand `DISTANCE_MAX`. This operand makes it possible to give the maximum distance which one authorizes between the node and the element nearest.

If the node does not answer the criterion of proximity the field will not be project on this node (i.e the node will not carry any component).

There is no value by default for `DISTANCE_MAX`. What wants to say that by default, the field will be prolonged apart from the matter also far one will need it.

4.12.3.4 CENTER / ANGL_NAUT / TRAN

These operands make it possible to define the geometrical transformation (rotation and/or translation) making it possible to pass from the face main slave to the face. The order carries out initially rotation then the translation.

Caution: the transformation is in the direction slave-Master.

This boundary condition applies to plane modelings ('PLAN' or 'AXIS') or voluminal ('3D').

4.13 Keyword ECHANGE_PAROI

4.13.1 Goal

Keyword factor usable to apply conditions of heat exchange between 2 walls. These walls can be defined in two manners:

- each one separately, by one or more meshes, or one or more groups of meshes (case of walls with a grid);
- starting from a list of cracks X-FEM, two walls corresponding then to the two lips of each crack. In this case it is also possible to impose the continuity of the field of temperature through the lips of each crack.

4.13.2 Syntax

- for `AFFE_CHAR_THER`

```
ECHANGE_PAROI=_F (
    # if the wall is with a grid
    ◆ / GROUP_MA_1 =          lgma,      [l_gr_maille]
      / MAILLE_1 =          lma,        [l_maille]
    ◆ / GROUP_MA_2 =          lgma,      [l_gr_maille]
      / MAILLE_2 =          lma,        [l_maille]
    ◆ COEF_H =              H,          [R]
    ◇ TRAN =                Lr,        [l_R]
```

if the wall is defined with cracks X-FEM

```

    ♦      CRACK =                lfiss,      [l_fiss_xfem]
    ♦ /    COEF_H =                H,         [R]
    /      TEMP_CONTINUE =        `YES'
)

• for AFFE_CHAR_THER_F

  ECHANGE_PAROI=_F (

    # if the wall is with a grid
    ♦ /    GROUP_MA_1 =            lgma,      [l_gr_maille]
    /      MAILLE_1 =              lma,      [l_maille]
    ♦ /    GROUP_MA_2 =            lgma,      [l_gr_maille]
    /      MAILLE_2 =              lma,      [l_maille]
    ♦      COEF_H =                H,         [function]
    ◊      TRAN =                  Lr,        [l_R]

    # if the wall is defined with cracks X-FEM
    ♦      CRACK =                lfiss,      [l_fiss_xfem]
    ♦ /    COEF_H =                H,         [function]
    /      TEMP_CONTINUE =        `YES'
  )

```

4.13.3 Operands

4.13.3.1 Case of walls with a grid

- ◆ COEF_H
 Coefficient of exchange enters the 2 walls.
 Reality for the operator AFFE_CHAR_THER, function for the operator AFFE_CHAR_THER_F.
- ◆ / GROUP_MA_1
 / MAILLE_1
- ◆ / GROUP_MA_2
 / MAILLE_2

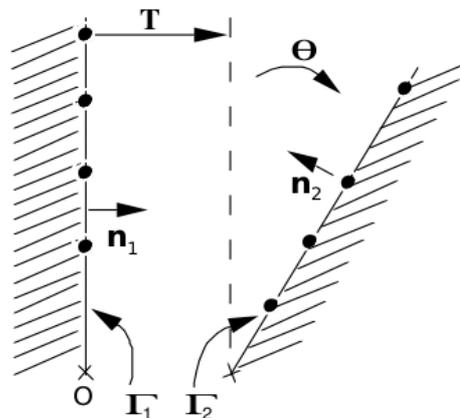


Figure 4.13.3-a

These operands make it possible to define the 2 lists of meshes representing for the subscripted list “_1” the wall Γ_1 for the subscripted list “_2” the wall Γ_2 .

The walls are in correspondence and must comprise the same number of meshes and nodes.

The limiting condition applied between these 2 walls is:

$$\text{on } \Gamma_1 : k \frac{\partial T_1}{\partial n_1} = h(T_2 - T_1) \quad n_1 \text{ normal external with } \Gamma_1$$

$$\text{on } \Gamma_2 : k \frac{\partial T_2}{\partial n_2} = h(T_1 - T_2) \quad n_2 \text{ normal external with } \Gamma_2$$

$$\text{where } \begin{aligned} T_1|_{\Gamma_1} &= T & T_2|_{\Gamma_1} &= T \circ f \\ T_1|_{\Gamma_2} &= T \circ f^{-1} & T_2|_{\Gamma_2} &= T \end{aligned}$$

f representing the bijection which puts in opposite a node of Γ_1 and a node of Γ_2 .

- ◆ TRAN
 Components of the vector translation.
 This operand makes it possible to define a virtual transformation (translation) approximate of Γ_1 in Γ_2 in order to ensure the bijectivity of the function out of screw - - screw. TRAN characterize a translation:

in 2D one thus has : TRAN = (tx, ty)
in 3D one has : TRAN = (tx, ty, tz)

4.13.3.2 Case of one or more cracks X-FEM

◆ CRACK

List of cracks X-FEM.

For each crack given in this list, the two walls in opposite are defined like the upper lip and the lower lip of the crack. There is thus as much couple of walls in opposite than of cracks present in this list.

◆ / COEF_H

Coefficient of exchange enters the lips of the crack. Reality for the operator AFFE_CHAR_THER, function for the operator AFFE_CHAR_THER_F.

If this keyword is present, the condition of exchange imposed between the lips of each crack given under the keyword CRACK is same as in the case of wall with a grid (see § preceding), while taking identitié for the bijection f since the upper lips and lower of each crack are geometrically confused.

/ TEMP_CONTINUE

This keyword can be indicated only with the value 'YES'.

If this keyword is present, one imposes the continuity of the field of temperature through the lips of each crack given under the keyword CRACK, by cancelling the degrees of freedom of enrichment ("Heaviside" and "ace-tip").

4.13.4 Use of ECHANGE_PAROI

In the case of a wall with a grid:

The user gives two lists of meshes from which the couples from paired nodes will result. These lists are initially sorted by type of mesh: the paired nodes will come from meshes of the identical type. For each mesh of the first list, one determines the mesh nearest in the second list by calculating all the distances from the nodes taken two to two (one traverses all the possible permutations). The distance minimum obtained defines at the same time the mesh in opposite and the couples of nodes paired for the two meshes concerned. As in LIAISON_GROUP [§4.11], it is possible to carry out a virtual geometrical transformation (rotation and/or translation) before calculating the distances.

In the case of one or several cracks X-FEM:

One defines as many couples of walls in opposite than there are cracks in the list indicated for the keyword CRACK.

- If the keyword COEF_H is present, all the cracks contained in this list will see affected the same coefficient of exchange (real or function). To assign a value of the clean coefficient of exchange to each crack, it is necessary to use the character répétable keyword factor ECHANGE_PAROI.
- If the keyword TEMP_CONTINUE is present, one imposes the continuity of the field of temperature through all the cracks contained in this list.

4.13.5 Meshes and modelings supporting this loading:

In the case of walls with a grid:

The lists of meshes given by the user must be made up of meshes of edge, of the meshes of coupling then are automatically generated. The table below provides a summary of the meshes of edges as of modelings for which this kind of loading is supported.

Mesh of edge	Modeling	Mesh of coupling generated
--------------	----------	----------------------------

Warning : The translation process used on this website is a "Machine Translation". It may be imprecise and inaccurate in whole or in part and is provided as a convenience.

Copyright 2017 EDF R&D - Licensed under the terms of the GNU FDL (<http://www.gnu.org/copyleft/fdl.html>)

SEG2, SEG3	PLAN, PLAN_DIAG AXIS, AXIS_DIAG	SEG22, SEG33
TRIA3, TRIA6, QUAD4, QUAD8, QUAD9	3D, 3D_DIAG	TRIA33, TRIA66, QUAD44, QUAD88, QUAD99

In the case of one or several cracks X-FEM:

The cracks must be created with the operator beforehand `DEFI_FISS_XFEM` [U4.82.08], and the model (well informed for the keyword `MODEL`) with the operator `MODI_MODELE_XFEM` [U4.41.11], which is not possible that in the following cases:

- modeling 3D for the meshes support HEXA8, PENTA6, PYRA5, TETRA4
- modelings PLAN and AXI for the meshes support QUAD4 and TRIA3

4.14 Keyword LIAISON_UNIF

4.14.1 Goal

Keyword factor allowing to impose the same value (unknown) on the temperatures of a set of nodes.

These nodes are defined by the groups of meshes, the meshes, the groups of nodes or the list of nodes to which they belong.

4.14.2 Syntax

- for AFFE_CHAR_THER and AFFE_CHAR_THER_F

```

LIAISON_UNIF =_F (
    ♦ / MESH = lma , [l_maille]
      / GROUP_MA = lgma, [l_gr_maille]
      / NODE = lno, [l_noeud]
      / GROUP_NO = lgno, [l_gr_noeud]

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

```

4.14.3 Operands

- ♦ / MESH
- ♦ / GROUP_MA
- ♦ / NODE
- ♦ / GROUP_NO

These operands make it possible to define a list of n nodes N_i from which one eliminated the redundancies (for MESH and GROUP_MA, it is connectivities of the meshes).

- ♦ DDL

This operand makes it possible to define a list of degrees of freedom $T_i (i=1, r)$ of r texts taken among: `TEMP', `TEMP_MIL', `TEMP_INF', `TEMP_SUP'.

$r \times (n-1)$ conditions 'kinematics' resulting are:

$$T_i(N_1) = T_i(N_k) \text{ for } k \in (2, \dots, n), i \in (1, \dots, r)$$

Note:

Components `TEMP_MIL', `TEMP_SUP', `TEMP_INF' can intervene only for nodes of elements of hull.

4.15 Keyword LIAISON_CHAMNO

4.15.1 Goal

Keyword factor usable to define a linear relation between all the temperatures present in a concept CHAM_NO.

4.15.2 Syntax

```
LIAISON_CHAMNO = _F (
    ♦ CHAM_NO = chamno, [cham_no]
    ♦ COEF_IMPO = B, [R]
    ♦ NUME_LAGR = / 'NORMAL', [DEFECT]
                / 'AFTER',
)
```

4.15.3 Operands

CHAM_NO =

Name of `chamno` who is used to define the linear relation. The temperatures connected are all those present in `chamno`. The coefficients to be applied to the temperatures are the values of these temperatures in `chamno`.

Example:

Let us suppose that one has one `chamno` relating to 3 nodes of names `NO1`, `NO2` and `NO3`.

Let us suppose that the values of the temperatures in these 3 nodes in `chamno` are respectively 2., 5.4 and 9.1. The linear relation that one will impose is:

$$2.T(NO1)+5.4T(NO2)+9.1T(NO3)=\beta$$

COEF_IMPO =

It is the value of the real coefficient β with the second member of the linear relation.

NUME_LAGR =

If `'NORMAL'`, the 2 multipliers of Lagrange associated with the relation will be such as the first will be located before all the terms implied in the relation and the second after, in the assembled matrix.

If `'AFTER'`, the 2 multipliers of Lagrange associated with the relation will be located after all the terms implied in the relation, in the assembled matrix.

This choice has the advantage of having an assembled matrix whose obstruction is weaker but has the disadvantage to be able to reveal a singularity in the matrix.

4.16 Keyword CONVECTION

4.16.1 Goal

Keyword usable to take into account the term of transport of heat by convection whose expression is $\rho C_p \cdot V \text{ grad } T$, appearing in the expression of the particulate derivative

$$\rho C_p \frac{dT}{dt} : \rho C_p \frac{dT}{dt} = \rho C_p \frac{\partial T}{\partial t} + \rho C_p V \text{ grad } T.$$

In the case of a liquid medium, V indicate the speed imposed of the fluid particle on the current point.

In the case of a mobile solid medium, V indicate the speed of the solid. In all the cases, it is supposed that the field speed is known a priori. The case of a mobile solid is rather frequent in practice. It relates to in particular the applications of welding or surface treatment which bring into play a source of heat moving in a given direction and at a speed.

The thermal problem is then studied in a reference frame related to the source (cf. THER_NON_LINE_MO [U4.54.03]).

4.16.2 Syntax

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

4.16.3 Operand

For AFFE_CHAR_THER and AFFE_CHAR_THER_F,

SPEED =

Name of the field speed at the moment when calculation is carried out.

This field is a concept `cham_no` of type `cham_no_depl_r`. It must have been defined on all the model for which one carries out calculation.

4.17 Keyword SOUR_NL

4.17.1 Goal

Keyword factor usable to apply **voluminal sources depending on the temperature** (2D or 3D) with one **field** defined by one or more meshes or of the groups of meshes of the type **voluminal**.

This loading is available only in the order AFFE_CHAR_THER_F. The values are provided via a tabulée function of the temperature of the type `function`. The source of heat, argument of the keyword SOUR, is a function of the temperature, other than any other parameter. Moreover, it is necessarily of a tabulée function and not about a formula.

4.17.2 Syntax

```
SOUR_NL=_F (
    ♦ / ALL = 'YES',
      / | MESH = lma, [l_maille]
        | GROUP_MA = lgma, [l_gr_maille]
    ♦ SOUR = sf, [function]
)
```

4.17.3 Operands

This loading applies to the types of meshes and following modelings:

Mesh	Modeling
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,

Value of the source depending on the temperature and presumedly constant on the element.

4.18 Keyword EVOL_CHAR

EVOL_CHAR = evch

The keyword factor EVOL_CHAR is usable to apply evolutionary loadings in the time of the type `evol_char` products by LIRE_RESU [U7.02.01] and containing fields of coefficients corresponding to a loading of the type EXCHANGE (see §4.7): the field T_EXT corresponds to the parameter TEMP_EXT and field COEF_H corresponds to the parameter COEF_H.