Operators

**AFFE_CHAR_MECA**

**AFFE_CHAR_MECA_C**

**AFFE_CHAR_MECA_F**

## 1 Goal

To affect loadings and boundary conditions on a mechanical model.

- For **AFFE_CHAR_MECA**, the affected values do not depend on any parameter and are defined by actual values.
- For **AFFE_CHAR_MECA_C**, the affected values do not depend on any parameter and are defined by complex values.
- For **AFFE_CHAR_MECA_F**, the affected values are function of one or more parameters as a whole \{INST, X, Y, Z, XF, YF, ZF\}.
Contents

1 Goal .......................................................................................................................................................... 1

2 General syntax ........................................................................................................................................ 5

3 General information ............................................................................................................................... 8

3.1 Principles ......................................................................................................................................... 8

3.2 Assumptions and limitations ........................................................................................................... 8

3.2.1 Linearity of the relations kinematics ......................................................................................... 8

3.2.2 Loadings of Neumann ................................................................................................................. 8

3.3 Availability of the loadings according to the type .......................................................................... 9

3.4 Possible error messages ................................................................................................................ 10

3.5 Choice of the units .......................................................................................................................... 11

3.6 Case of the great transformations .................................................................................................. 11

3.6.1 Problems ................................................................................................................................... 11

3.6.2 Following loadings .................................................................................................................... 12

3.7 Designation of the topological entities of assignment of the loadings .......................................... 12

3.8 Rules of overload and remanence ................................................................................................. 12

3.9 Definition of the reference marks ................................................................................................ 13

3.9.1 Normals and tangents with the meshes .................................................................................. 13

3.9.2 Case of the elements of structure ........................................................................................... 14

3.9.3 Definition of a reference mark by the nautical angles ............................................................... 14

4 Operands generals .................................................................................................................................. 16

4.1 Operand MODEL ............................................................................................................................ 16

4.2 Operand DOUBLE_LAGRANGE ................................................................................................. 16

4.3 Operand VERI_NORM .................................................................................................................... 16

4.4 Operand NUME_LAGR ................................................................................................................... 16

4.5 Operand INFORMATION ............................................................................................................... 16

4.6 Operand ANGL_NAUT/CENTRE/TRAN .................................................................................... 16

4.7 Operand LIAISON_EPX ................................................................................................................ 17

5 Loadings of the Dirichlet type ............................................................................................................. 18

5.1 Degrees of freedom ......................................................................................................................... 18

5.2 Conflicts between the degrees of freedom ..................................................................................... 20

5.3 Operations of pairing ....................................................................................................................... 20

5.3.1 Pairing node with node (compatible grids) ............................................................................. 20

5.3.2 Pairing mesh-with-node (incompatible grids) ........................................................................... 21

5.4 Keyword DDL_IMPO ..................................................................................................................... 21

5.5 Keyword ARETE_IMPO ............................................................................................................... 22

5.6 Keyword FACE_IMPO ................................................................................................................... 23

5.7 Keyword LIAISON_DDL .............................................................................................................. 24

5.8 Keyword LIAISON_OBLIQUE ..................................................................................................... 25

5.9 Keyword LIAISON_UNIF ............................................................................................................. 25
5.10 Keyword LIAISON_CHAMNO
5.11 Keyword CHAMNO_IMPO
5.12 Keyword LIAISON_GROUP
5.13 Keyword LIAISON_MAIL
5.13.1 Use and examples
5.13.2 Some remarks and precautions of use
5.13.3 Keywords
5.14 Keyword LIAISON_PROJ
5.14.1 Goal
5.14.2 Syntax (AFFE_CHAR_MECA only)
5.14.3 Operands
5.15 Keyword LIAISON_CYCL
5.16 Keyword LIAISON_SOLIDE
5.17 Keyword LIAISON_ELEM
5.17.1 Option ‘3D_POU’
5.17.2 Option ‘3D_POU_ARLEQUIN’
5.17.3 Option ‘2D_POU’
5.17.4 Option ‘COQ_POU’
5.17.5 Option ‘3D_TUYAU’
5.17.6 Option ‘COQ_TUYAU’
5.17.7 Option ‘PLAQ_POUT_ORTH’
5.18 Keyword LIAISON_RBE3

6 Loadings of the Dirichlet type for the elements of structure
6.1 Keyword DDL_POUTRE
6.2 Keyword LIAISON_COQUE

7 Loadings of the Neumann type
7.1 Keyword FORCE_NODALE
7.2 Keyword FORCE_ARETE
7.3 Keyword FORCE_CONTOUR
7.4 Keyword FORCE_FACE
7.5 Keyword FORCE_INTERNE
7.6 Keyword PRES_REP
7.7 Keyword EVOL_CHAR
7.8 Keyword EFFE_FOND
7.9 Keyword GRAVITY
7.10 Keyword ROTATION
7.11 Keyword PRE_SIGM
7.12 Keyword PRE_EPSI
7.13 Keyword FORCE_ELEC
7.14 Keyword INTE_ELEC
7.15 Keyword VECT_ASSE

8 Loadings of the Neumann type for the elements of structure

8.1 Keyword FORCE_POUTRE

8.2 Keyword FORCE_TUYAU

8.3 Keyword FORCE_COQUE

9 Other loadings

9.1 Keyword LIAISON_INTERF

9.2 Keyword RELA_CINE_BP

9.3 Keyword IMPE_FACE

9.4 Keyword VITE_FACE

9.5 Keyword ONDE_PLANE

9.6 Keyword ONDE_FLUI

9.7 Keyword FLUX_THM_REP

9.8 Keyword FORCE_SOL
# General syntax

```plaintext
ccharg [char_meca] = AFFE_CHAR_MECA(
  ♦ MODEL = Mo
  ◊ INFORMATION = [1, 2]
)[defect]
  ◊ DOUBLE_LAGRANGE = 'YES', 'NOT'
  ◊ WORMI_NORM = 'YES', 'NOT'
  ◊ LIAISON_EFX = 'YES'
  ♦ | GRAVITY = to see keyword GRAVITY
  | ROTATION = to see keyword ROTATION
  | DDL_IMPO = to see keyword DDL_IMPO
  | DDL_POUTRE = to see keyword DDL_POUTRE
  | FACE_IMPO = to see keyword FACE_IMPO
  | CHAMNO_IMPO = to see keyword CHAMNO_IMPO
  | ARETE_IMPO = to see keyword ARETE_IMPO
  | LIAISON_DDL = to see keyword LIAISON_DDL
  | LIAISON_OBLIQUE = to see keyword LIAISON_OBLIQUE
  | LIAISON_GROUP = to see keyword LIAISON_GROUP
  | LIAISON_MAIL = to see keyword LIAISON_MAIL
  | LIAISON_PROJ = to see keyword LIAISON_PROJ
  | LIAISON_CYCL = to see keyword LIAISON_CYCL
  | LIAISON_SOLIDE = to see keyword LIAISON_SOLIDE
  | LIAISON_ELEM = to see keyword LIAISON_ELEM
  | LIAISON_UNIF = to see keyword LIAISON_UNIF
  | LIAISON_CHAMNO = to see keyword LIAISON_CHAMNO
  | LIAISON_RBE3 = to see keyword LIAISON_RBE3
  | LIAISON_INTERF = to see keyword LIAISON_INTERF
  | VECT_ASSE = to see keyword VECT_ASSE
)[§ 7.9]
  | | FORCE_NODALE = to see keyword FORCE_NODALE
  | | FORCE_FACE = to see keyword FORCE_FACE
  | | FORCE_ARETE = to see keyword FORCE_ARETE
  | | FORCE_CONTOUR = to see keyword FORCE_CONTOUR
  | | FORCE_INTERNE = to see keyword FORCE_INTERNE
  | | FORCE_POUTRE = to see keyword FORCE_POUTRE
  | | FORCE_TUYAU = to see keyword FORCE_TUYAU
  | | FORCE_COQUE = to see keyword FORCE_COQUE
  | | LIAISON_COQUE = to see keyword LIAISON_COQUE
)[§ 7.5]
  | | PRE_SIGM = to see keyword PRE_SIGM
  | | PRES_REP = to see keyword PRES_REP
  | | EFFE_FOND = to see keyword EFFE_FOND
  | | PRE_EPSI = to see keyword PRE_EPSI
  | | FORCE_POUTRE = to see keyword FORCE_POUTRE
  | | FORCE_TUYAU = to see keyword FORCE_TUYAU
  | | FORCE_COQUE = to see keyword FORCE_COQUE
  | | LIAISON_COQUE = to see keyword LIAISON_COQUE
)[§ 7.8]
)```

---

_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 2020 EDF R&D - Licensed under the terms of the GNU FDL (http://www.gnu.org/copyleft/fdl.html)_
charg [char_meca] = AFFE_CHAR_MECA_C(
    MODEL = Mo, [model]
)[defect]
    INFORMATION = [1, 2]
[$5.4]
    DDL_IMPO = to see keyword DDL_IMPO
[$5.7]
    LIAISON_DDL = to see keyword LIAISON_DDL
    FORCE_POUTRE = to see keyword FORCE_POUTRE

charg [char_meca] = AFFE_CHAR_MECA_F(
    MODEL = Mo, [model]
)[defect]
    INFORMATION = [1, 2]
[$7.1]
    WORMI_NORM = '/YES', / 'NOT'
[$5.4]
    FACE_IMPO = to see keyword FACE_IMPO
[$5.6]
    LIAISON_DDL = to see keyword LIAISON_DDL
[$5.7]
    LIAISON_OBLIQUE = to see keyword LIAISON_OBLIQUE
[$5.8]
    LIAISON_GROUP = to see keyword LIAISON_GROUP
[$5.9]
    FORCE_NODALE = to see keyword FORCE_NODALE

[$7.4]
    FORCE_FACE = to see keyword FORCE_FACE

[$7.2]
    FORCE_ARETE = to see keyword FORCE_ARETE

[$7.3]
    FORCE_CONTOUR = to see keyword FORCE_CONTOUR

[$7.5]
    FORCE_INTERNE = to see keyword FORCE_INTERNE

[$7.6]
    PRES_REP = to see keyword PRES_REP

[$7.8]
    EFFE_FOND = to see keyword EFFE_FOND

[$7.12]
    PRE_EPSI = to see keyword PRE_EPSI

[$8.1]
    FORCE_POUTRE = to see keyword FORCE_POUTRE

[$8.2]
    FORCE_TUYAU = to see keyword FORCE_TUYAU
FORCE_COQUE = to see keyword FORCE_COQUE

LIAISON_COQUE = to see keyword LIAISON_COQUE

IMPE_FACE = to see keyword IMPE_FACE

VITE_FACE = to see keyword VITE_FACE

ONDE_PLANE = to see keyword ONDE_PLANE

FLUX_THM_REP = to see keyword FLUX_THM_REP
3 General information

3.1 Principles

There are three main categories of operand:

- Operands applying of the conditions kinematics or loadings of Dirichlet, i.e. relations between the degrees of freedom. In `AFFE_CHAR_MECA`, these conditions are applied by dualisation (method of the Lagrange doubles, to see [R2.03.01]);
- Operands applying of the loadings of the type “forces” or loadings of Neumann, applied in weak form, which implies the use of a digital diagram of integration. Certain loadings imply the presence of elements of edge in the model;
- Operands applying of the special loadings, including mixed type Dirichlet/Neumann.

Most operands are built on the same principle:

- Specification of the place of application of the limiting conditions by the standard keywords `GROUP_NO`, `GROUP_MA` and sometimes `SANS_GROUP_NO` and `SANS_GROUP_MA`.
- Specification of the affected components, which are divided into three groups:
  - Components standards of the size considered. It is the size `DEPL_R` (or `DEPL_C` or `DEPL_F`), representing the degrees of freedom of the problem of mechanics (see § 5.1);
  - Combined components `DNOR` and `DTAN`, which builds a combination between the components of the size `DEPL_R` on considerations relating to the tangents and the normal;
  - Components in efforts, moments or pressure using is the size `FORC_R` (or `FORC_C` or `FORC_F`), size `PRES_R` (or `PRES_C` or `PRES_F`);
- The affected components must be of the good type according to the operator used:
  - Type reality for the operator `AFFE_CHAR_MECA`;
  - Type complex for the operator `AFFE_CHAR_MECA_C`;
  - Type function (created in particular by one of the operators `DEFI_FONCTION`, `DEFI_NAPPE` or `DEFI_CONSTANTE`) for the operator `AFFE_CHAR_MECA_F`. This is true near with an exception: the argument of `COEF_MULT` for the keyword factor `LIAISON_DDL` in `AFFE_CHAR_MECA_F` is obligatorily of real type.

3.2 Assumptions and limitations

Besides the definition, specific assumptions and limitations to each loading, there exist general assumptions which one will point out here.

3.2.1 Linearity of the relations kinematics

It is pointed out that a kinematic relation makes it possible to write an equation of the type:

\[ \sum_{i=1}^{r} \alpha_i U_i = \beta \]  

(1)

With \( U_i \) the list of \( r \) Degrés of freedom, \( \alpha_i \) coefficients and \( \beta \) the second member.

It should be noted that the relations kinematics of `Code_Aster` are relations linear, i.e.:

- They cannot depend a priori deformation or movement of the structure: they remain valid only on the assumption of the small disturbances, except contrary mention (see `LIAISON_SOLID`).
- Coefficients \( \alpha_i \) linear relation cannot be functions of time, because the matrix \( B \) conditions of Dirichlet is constant during all the transient. On the other hand, they can be functions of the geometry initial;
- The second member \( \beta \) can be a function of time or geometry initial.

3.2.2 Loadings of Neumann

Contrary to the conditions kinematics, it is completely possible that some loadings of Neumann are non-linear, and, in particular, depend on the structural deformation. Such loadings are commonly
called loadings **follower**. Nevertheless, in this case, the problem becoming non-linear, it is necessary to use an operator of adequate calculation like STAT_NON_LINE and DYNA_NON_LINE and to specify that these loadings are indeed considered like follower (see [U4.51.03]).

Most loadings of Neumann (except FORCE_NODALE) are applied in weak form, i.e. one uses a digital formula of squaring. Moreover, one cannot simultaneously apply a loading of Neumann and a loading of Dirichlet to the same node and in the same direction. So it can exist a difference between the theoretical solution and the solution finite elements.

For example, on a structure insufficiently with a grid, it is possible to note a difference between the sum of the nodal efforts corresponding to the loading of gravity and the value of the actual weight, the corresponding variation *roughly speaking* with the number of embedded nodes of the structure.

A refinement of the grid makes it possible to minimize this difference. One can also to make so that the finite elements, on which conditions kinematics are imposed, are of a sufficiently small size so that their weight is negligible in front of that of the total structure. Another solution is to duplicate the nodes on which the kinematic condition is imposed and to do for example one LIAISON_DDL between the two nodes or to use discrete elements.

### 3.3 Availabilities of the loadings according to the type

The loadings available are not inevitably applicable in the three operators AFFE_CHAR_MECA. Here the list of the availabilities according to the type of the operator:

<table>
<thead>
<tr>
<th>Keyword</th>
<th>AFFE_CHAR_MECA</th>
<th>AFFE_CHAR_MECA_C</th>
<th>AFFE_CHAR_MECA_F</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARETE_IMPO</td>
<td>YES</td>
<td>NOT</td>
<td>NOT</td>
</tr>
<tr>
<td>CHAMNO_IMPO</td>
<td>YES</td>
<td>NOT</td>
<td>NOT</td>
</tr>
<tr>
<td>DDL_IMPO</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>EFFE_FOND</td>
<td>YES</td>
<td>NOT</td>
<td>YES</td>
</tr>
<tr>
<td>EVOL_CHAR</td>
<td>YES</td>
<td>NOT</td>
<td>NOT</td>
</tr>
<tr>
<td>FACE_IMPO</td>
<td>YES</td>
<td>NOT</td>
<td>YES</td>
</tr>
<tr>
<td>FLUX_THM_REP</td>
<td>YES</td>
<td>NOT</td>
<td>YES</td>
</tr>
<tr>
<td>FORCE_ARETE</td>
<td>YES</td>
<td>NOT</td>
<td>YES</td>
</tr>
<tr>
<td>FORCE_CONTOUR</td>
<td>YES</td>
<td>NOT</td>
<td>YES</td>
</tr>
<tr>
<td>FORCE_COQUE</td>
<td>YES</td>
<td>NOT</td>
<td>YES</td>
</tr>
<tr>
<td>FORCE_ELEC</td>
<td>YES</td>
<td>NOT</td>
<td>NOT</td>
</tr>
<tr>
<td>FORCE_FACE</td>
<td>YES</td>
<td>NOT</td>
<td>YES</td>
</tr>
<tr>
<td>FORCE_INTERNE</td>
<td>YES</td>
<td>NOT</td>
<td>YES</td>
</tr>
<tr>
<td>FORCE_NODALE</td>
<td>YES</td>
<td>NOT</td>
<td>YES</td>
</tr>
<tr>
<td>FORCE_POUTRE</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>FORCE_SOL</td>
<td>YES</td>
<td>NOT</td>
<td>NOT</td>
</tr>
<tr>
<td>FORCE_TUYAU</td>
<td>YES</td>
<td>NOT</td>
<td>YES</td>
</tr>
<tr>
<td>IMPE_FACE</td>
<td>YES</td>
<td>NOT</td>
<td>YES</td>
</tr>
<tr>
<td>INTE_ELEC</td>
<td>YES</td>
<td>NOT</td>
<td>NOT</td>
</tr>
<tr>
<td>LIAISON_CHAMNO</td>
<td>YES</td>
<td>NOT</td>
<td>NOT</td>
</tr>
<tr>
<td>LIAISON_CYCL</td>
<td>YES</td>
<td>NOT</td>
<td>NOT</td>
</tr>
<tr>
<td>LIAISON_DDL</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
</tbody>
</table>
3.4 Possible error messages

It happens sometimes that a mechanical ordering of calculation stops in fatal error during the calculation of the second elementary members due to the loadings defined in the orders `AFFE_CHAR_MECA_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 establishes a correspondence between keywords factors and the names of option of calculation which they activate:

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Name of the option</th>
</tr>
</thead>
<tbody>
<tr>
<td>EVOL_CHAR</td>
<td>CHAR_MECA_PRES_R</td>
</tr>
<tr>
<td></td>
<td>CHAR_MECA_FR3D3D</td>
</tr>
<tr>
<td></td>
<td>CHAR_MECA_FR2D2D</td>
</tr>
<tr>
<td></td>
<td>CHAR_MECA_FR2D3D</td>
</tr>
<tr>
<td></td>
<td>CHAR_MECA_FR1D2D</td>
</tr>
<tr>
<td>GRAVITY</td>
<td>CHAR_MECA_PESA_R</td>
</tr>
<tr>
<td>ROTATION</td>
<td>CHAR_MECA_ROTA_R</td>
</tr>
<tr>
<td>PRE_SIGM</td>
<td>FORC_NODA</td>
</tr>
<tr>
<td>FORCE_NODALE</td>
<td>CHAR_MECA_FORC_R</td>
</tr>
<tr>
<td></td>
<td>CHAR_MECA_FORC_F</td>
</tr>
<tr>
<td>FORCE_ARETE</td>
<td>CHAR_MECA_FR1D3D</td>
</tr>
<tr>
<td></td>
<td>CHAR_MECA_FF1D3D</td>
</tr>
<tr>
<td>FORCE_CONTOUR</td>
<td>CHAR_MECA_FR1D2D</td>
</tr>
<tr>
<td></td>
<td>CHAR_MECA_FF1D2D</td>
</tr>
<tr>
<td>FORCE_FACE</td>
<td>CHAR_MECA_FR2D3D</td>
</tr>
<tr>
<td></td>
<td>CHAR_MECA_FF2D3D</td>
</tr>
</tbody>
</table>
3.5 Choice of the units

For the loadings of Neumann, the forces are to be provided per unit of grid for the linear efforts, per unit of grid squared for the surface efforts and per unit of grid to the cube for the voluminal efforts), in coherence with the definition of the properties materials (Young modulus for example). In the case axisymmetric, the forces required are brought back to a sector of $\frac{1}{2\pi}$ radian (to divide the real loading by $2\pi$).

3.6 Case of the great transformations

3.6.1 Problems

When one applies a loading of Dirichlet to a structure in great transformations (great displacements, great rotations), it is necessary to take care of the good taking into account of this one. I.e. that it should be taken care that this loading is applicable when the assumption of the small disturbances is not checked any more.

It is naturally the case of the following loadings:

- **DDL_IMPO** but if one uses **AFFE_CHAR_MECA_F**, it should be done with functions which represent imposed displacement **reality** (curvilinear in general). If it is not done, the interpolation in time makes us pass by intermediate states “false”;
- **LIAISON_MAIL** + **TYPE_RACCORD=' MASSIF';**
- **LIAISON_UNIF**;
- **LIAISON_OBLIQUE**;
- **LIAISON_DDL**
- **LIAISON_CHAM_NO**
- **CHAMNO_IMPO**

| FORCE_INTERNE         | CHAR_Meca_FR2D2D |
|                       | CHAR_Meca_FR3D3D |
|                       | CHAR_Meca_FF2D2D |
|                       | CHAR_Meca_FF3D3D |
| PRES_REP              | CHAR_Meca_PRES_R |
|                       | CHAR_Meca_PRES_F |
| EFFE_FOND             | CHAR_Meca_EFON_R |
|                       | CHAR_Meca_EFON_F |
| PRE_EPSI              | CHAR_Meca_EPSI_R |
|                       | CHAR_Meca_EPSI_F |
| FORCE_ELEC            | CHAR_Meca_FRELEC |
| INTE_ELEC             | CHAR_Meca_FRLAPL |
| FORCE_POUTRE          | CHAR_Meca_FR1D1D |
|                       | CHAR_Meca_FC1D1D |
|                       | CHAR_Meca_FF1D1D |
| FORCE_TUYAU           | CHAR_Meca_PRES_R |
|                       | CHAR_Meca_PRES_F |
| FORCE_COQUE           | CHAR_Meca_FRCO2D |
|                       | CHAR_Meca_FRCO3D |
|                       | CHAR_Meca_FFCO2D |
|                       | CHAR_Meca_FFCO3D |
| FLUX_THM_REP          | CHAR_Meca_FLUX_R |
|                       | CHAR_Meca_FLUX_F |
• LIAISON_SOLIDE on condition that applying the assumption of a following loading (see §3.6).

On the other hand, for the following loadings, the assumption of the great transformations is not applicable and will cause false results:
• FACE_IMPO with DNOR=f (T) : Lhas “normal” changes into great rotations;
• LIAISON_MAIL + TYPE_RACCORD='COQUE_MASSIF' and 'MASSIF_COQUE’;
• LIAISON_ELEM;
• LIAISON_RBE3.

3.6.2 Following loadings

For the non-linear operators (STAT_NON_LINE and DYNA_NON_LINE), certain loadings can be “following", i.e. their application depends on displacement and thus changes with each iteration of Newton. It is then necessary that the user specifies it by the operand TYPE_CHARGE in the keyword factor EXCIT of these orders (see [U4.51.03]). The fact of specifying that the loading is following addition sometimes a contribution in the matrix of rigidity (see for example [R3.03.04]) and can make it not-symmetrical. However, for the loading EVOL_CHAR (see §7.7), it is not necessary to specify that the loadings are following, they are to it automatically by default. The fact of specifying it simply will activate the additional matric contribution and will thus act on the speed of convergence (and not on the precision of the result).

3.7 Designation of the topological entities of assignment of the loadings

In a general way, when the entities on which values must be affected are defined:
• On only one node: the operand is used GROUP_NO who should obviously contain one node;
• On a list of nodes: one can useR the operand GROUP_NO but also the operand GROUP_MA or TOUT=' OUI' to affect on all the grid;
• On only one nets: the operand is used GROUP_MA who should obviously contain only one nets;
• On a list of meshs: one can use the operand GROUP_MA or TOUT=' OUI' to affect on all the grid;

Certain keywords need to define several topological entities (groups of nodes in opposite for example), in this case, the names can vary slightly (GROUP_NO_2, GROUP_MA_ESCL, etc). It is possible in most key words excluding from the nodes or the meshs using operator of the type SANS_*. This functionality avoids redefining groups in your grid or the order DEFI_GROUP.

3.8 Rules of overload and remanence

To define the field of assignment most simply possible, one uses the rule of overload defined in the document [U1.03.00] which one points out the principles:
• The assignments are done by superimposing the effects of the various loadings;
• In the event of conflict, the last loading overrides the precedents;

If for example, the made user:

FORCE_FACE= _F (GROUP_MA=' G1', FX=12.),
PRES_REP= _F (GROUP_MA=' G1', PRES=13.)

And if the normal for G1 is directed according to X, then all will occur as if one had made:

FORCE_FACE= _F (GROUP_MA=' G1', FX=25.)

The rule of preceding overload must be supplemented by another rule to specify what occurs when one can affect several quantities for each occurrence of one loading. That is to say for example:

FORCE_INTERNE= (}
The rule of overload tells us that the second occurrence of \texttt{FORCE\_INTERNE} overload the first. But what is worth \( FX \) on a mesh belonging to \texttt{GM1}? Was it erased by the second occurrence? If the only rule of overload is observed, \( FX \) is not defined on \texttt{GM1}. One thus uses a second rule known as of remanence which specifies that during the application of the rule of overload on occurrences, one preserves the components which are not overloaded. By applying the rule of remanence to the example, \( FX \) preserve the affected value as a preliminary. All the elements of the model thus have a value for \( FX \) and elements of \texttt{GM1} have a value at the same time for \( FX \) and \( FY \).

### 3.9 Definition of the reference marks

Most loadings are defined in the reference mark \texttt{total} grid, except:

- For the elements of structure (§3.9.2);
- For the keywords \texttt{DNOR}, \texttt{DTAN} and loadings of type pressure. In this case, it is necessary to define normal and tangent (see §3.9.1), even if required to use the order \texttt{MODI\_MAILLAGE};
- When the keyword \texttt{ANGL\_NAUT} is usable (see §3.9.3);

In the other cases, it is generally possible to define functions of space (subject remaining within the framework of the assumptions of the §3.2).

#### 3.9.1 Normals and tangents with the meshes

One gives here the standard definition of the normals and the tangents according to the type of mesh of edge:

- For the elements segments in 2D, the tangent is that defined by the segment directed by its first two nodes, the normal \( n \) is then such as \((n,t)\) form a direct reference mark

![Diagram of a segment with its normal and tangent](image)

- For the elements triangles or quadrangles in 3D, \( L \) orientation of the normal \( n \) is that corresponding to the direct direction of the description of the mesh.

![Diagram of a triangle and a quadrangle with their normals](image)

If \texttt{DNOR} (or \texttt{DTAN}) is specified, the normal (or the tangent) on a node is the average of the normals or the tangents of the meshes on which are affected the limiting conditions and which have this joint node (except for the curved quadratic elements where the normal is correctly calculated in any point).

The operator \texttt{MODI\_MAILLAGE} allows to make sure of the continuity of the orientation of the normal at the edges of the solid elements of continuous medium.
3.9.2 Case of the elements of structure

The elements of structure (beams, plates and hulls) have their own reference mark room whose definition is given in the documentation of the order AFFE_CARA_ELEM [U4.42.01].

3.9.3 Definition of a reference mark by the nautical angles

Certain loadings give the opportunity of giving their direction of application by using nautical angles whose one points out the definition here.

Nautical angles $\alpha$, $\beta$, $\gamma$ provide in degrees, are the angles allowing to pass from the total reference mark of definition of the coordinates of the nodes $(P, X, Y, Z)$ with the local reference mark $(P_3, X_3, Y_3, Z_3)$. This one is obtained by three rotations:

- A rotation of angle $\alpha$ around $Z$, transforming $(X Y Z)$ in $(X_1 Y_1 Z_1)$ with $Z_1 \equiv Z$ [Figure 3.9.3-1];
- A rotation of angle $\beta$ around $Y_1$, transforming $(X_1 Y_1 Z_1)$ in $(X_2 Y_2 Z_2)$ with $Y_2 \equiv Y_1$ [Figure 3.9.3-2];
- A rotation of angle $\gamma$ around $X_2$, transforming $(X_2 Y_2 Z_2)$ in $(X_3 Y_3 Z_3)$ with $X_3 \equiv X_2$ [Figure 3.9.3-3];

![Figure 3.9.3-1: angle $\alpha$.](image1)

![Figure 3.9.3-2: angle $\beta$.](image2)

Note: for the figure 3.9.3-2, the swing angle $\beta$ is negative.

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 2020 EDF R&D - Licensed under the terms of the GNU FDL (http://www.gnu.org/copyleft/fdl.html)
Figure 3.9.3-3: angle $\gamma$.

The local reference mark is: $(X_3, Y_3, Z_3)$

Figure 3.9.3-4: Representation of the total and local reference mark.
4 Operands generals

4.1 Operand MODEL

- MODEL
The keyword MODEL a concept produced by the operator waits AFFE_MODELE where are defined the types of finite elements affected on the grid. The model is necessarily of the type MECHANICS.

4.2 Operand DOUBLE_LAGRANGE

- DOUBLE_LAGRANGE = ‘YES’/‘NOT’
This keyword makes it possible to say if the user or not wishes to duplicate the multipliers of Lagrange used to define dualiser the boundary conditions in the assembled matrix. Concretely, to duplicate the multipliers of Lagrange makes it possible to use linear solvers not allowing the swivelling. Not to duplicate Lagrange makes it possible to reduce the number of degree of freedom of the problem (and thus size of the problem to be solved) but its use is limited to solvers MUMPS and Petsc.

4.3 Operand VERI_NORM

- VERI_NORM = ‘YES’/‘NOT’
Checking of the orientation of the normals to the surface meshes in 3D (meshes of skin SORTED or QUAD) and linear in 2D (meshes of skin SEG). If a normal is not outgoing, there is emission of an error message fatal.

To reorientate the meshes in order to have outgoing normals, the operator should be used MODI_MAILLAGE [U4.23.04] keyword ORIE_PEAU_2D and ORIE_PEAU_3D.

No checking is made on the hulls. To check their orientation, one also returns to the operator MODI_MAILLAGE keyword ORIE_NORM_COQUE.

4.4 Operand NUME_LAGR

- NUME_LAGR = ‘NORMAL’/‘AFTER’
This keyword is useful in the imposition of complex conditions kinematics. If NUME_LAGR = ‘NORMAL’, then the two 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 NUME_LAGR = ‘APRES’, the two 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 myhas the disadvantage to be able to reveal a singularity in the matrix.

4.5 Operand INFORMATION

- INFORMATION
Level of the impressions on the file message.

4.6 Operand ANGL_NAUT/CENTRE/TRAN

- ANGL_NAUT = (has, B, c)
Allows to define a reference mark by the nautical angles in degrees (see § 3.9.3 ).

- CENTER = (cx, cy, cz)
Coordonnées of the centre of rotation (in the total reference mark).
4.7 Operand LIAISON_EPX

◊ LIAISON_EPX =/'YES'

The presence of this keyword involve the creation of one table containing the list of the relations linear produced by the operator. This table is read again by CALC EUROPLEXUS (if the loading is provided to him). Only the keyword LIAISON_MAIL is compatible with LIAISON_EPX.

Note: there is ofother loadings authorized in CALC EUROPLEXUS but those have a direct equivalent among the features DE EUROPLEXUS. For the connections not having an equivalent, this functionality allows to transmit directly with EUROPLEXUS the list of the relations linear created by code_aster.
# 5 Loadings of the Dirichlet type

## 5.1 Degrees of freedom

The loadings of Dirichlet are imposed on the degrees of freedom of the size DEPL_R (or DEPL_C or DEPL_F), representing the degrees of freedom of the problem of mechanics (or of thermo-hydro-mechanics or hydraulics). We point out here the significance of these various degrees of freedom:

<table>
<thead>
<tr>
<th>Name</th>
<th>Modeling</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DX, DY, DZ</td>
<td>Very except 2D_FLUI_PESA</td>
<td>Components of displacement in translation in the total reference mark</td>
</tr>
<tr>
<td>DZ</td>
<td>2D_FLUI_PESA</td>
<td>Imposed displacement of free surface</td>
</tr>
<tr>
<td>DRX, DRY, MARTINI, DRZ</td>
<td>Discrete elements, of beam, hull or plate</td>
<td>Components of displacement in rotation in the total reference mark</td>
</tr>
<tr>
<td>GRX</td>
<td>POU_D_TG</td>
<td>Value of the warping of the beam</td>
</tr>
<tr>
<td>NEAR</td>
<td>3D_FLUIDE</td>
<td>Acoustic pressure in the fluid</td>
</tr>
<tr>
<td>NEAR</td>
<td>3D_JOINT_CT</td>
<td>Pressure of the interstitial fluid</td>
</tr>
<tr>
<td>NEAR</td>
<td>Formulation second gradient</td>
<td>Multiplier of Lagrange introduced for the mixed formulation</td>
</tr>
<tr>
<td>PHI</td>
<td>3D_FLUIDE, FLUI_STRU, 2D_FLUI_PESA</td>
<td>Potential of displacements of the fluid</td>
</tr>
<tr>
<td>TEMP</td>
<td>THM, THHM, THH</td>
<td>Temperature</td>
</tr>
<tr>
<td>PRE1</td>
<td>THM, THHM, THH, HM, HMM</td>
<td>Capillary pressure or pressure of the liquid or gas</td>
</tr>
<tr>
<td>PRE2</td>
<td>THM, THHM, THH</td>
<td>Pressure of gas</td>
</tr>
<tr>
<td>UIx</td>
<td>Pipes</td>
<td>Warping “in plane” of mode X ¹</td>
</tr>
<tr>
<td>VIx, WIx</td>
<td>Pipes</td>
<td>Ovalizations “in plane” of mode X</td>
</tr>
<tr>
<td>UOx</td>
<td>Pipes</td>
<td>Warping “out of planes” mode X</td>
</tr>
<tr>
<td>VOx, WOx</td>
<td>Pipes</td>
<td>Ovalizations “out of planes” mode X</td>
</tr>
<tr>
<td>W0, WI1, WO1</td>
<td>Pipes</td>
<td>Degrees of freedom of swelling and mode 1 on ovalization</td>
</tr>
<tr>
<td>GONF</td>
<td>Incompressible formulation _INCO_UPG</td>
<td>Swelling</td>
</tr>
</tbody>
</table>

¹ X is worth 2 and 3 for TUYAU_3M and TUYAU_6M, 4, 5 and 6 only for TUYAU_6M
<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>V11</td>
<td>Formulation second gradient</td>
</tr>
<tr>
<td>V12</td>
<td>Components of the tensor of deformation microscopic</td>
</tr>
<tr>
<td>V13</td>
<td></td>
</tr>
<tr>
<td>V21</td>
<td></td>
</tr>
<tr>
<td>V22</td>
<td></td>
</tr>
<tr>
<td>V23</td>
<td></td>
</tr>
<tr>
<td>V31</td>
<td></td>
</tr>
<tr>
<td>V32</td>
<td></td>
</tr>
<tr>
<td>V33</td>
<td></td>
</tr>
<tr>
<td>PRES11</td>
<td>Formulation second gradient</td>
</tr>
<tr>
<td>PRES12</td>
<td></td>
</tr>
<tr>
<td>PRES13</td>
<td></td>
</tr>
<tr>
<td>PRES21</td>
<td></td>
</tr>
<tr>
<td>PRES22</td>
<td></td>
</tr>
<tr>
<td>PRES23</td>
<td></td>
</tr>
<tr>
<td>PRES31</td>
<td></td>
</tr>
<tr>
<td>PRES32</td>
<td></td>
</tr>
<tr>
<td>PRES33</td>
<td></td>
</tr>
<tr>
<td>LAGS_C</td>
<td>Contact continuous or XFEM</td>
</tr>
<tr>
<td>LAGS_F1</td>
<td>Contact continuous or XFEM</td>
</tr>
<tr>
<td>LAGS_F2</td>
<td></td>
</tr>
<tr>
<td>H1X</td>
<td>XFEM</td>
</tr>
<tr>
<td>H1Y</td>
<td></td>
</tr>
<tr>
<td>H1Z</td>
<td></td>
</tr>
<tr>
<td>H2X</td>
<td></td>
</tr>
<tr>
<td>H2Y</td>
<td></td>
</tr>
<tr>
<td>H2Z</td>
<td></td>
</tr>
<tr>
<td>H3X</td>
<td></td>
</tr>
<tr>
<td>H3Y</td>
<td></td>
</tr>
<tr>
<td>H3Z</td>
<td></td>
</tr>
<tr>
<td>H4X</td>
<td></td>
</tr>
<tr>
<td>H4Y</td>
<td></td>
</tr>
<tr>
<td>H4Z</td>
<td></td>
</tr>
<tr>
<td>H1PRE1</td>
<td>HM-XFEM</td>
</tr>
<tr>
<td>H2PRE1</td>
<td></td>
</tr>
<tr>
<td>H3PRE1</td>
<td></td>
</tr>
<tr>
<td>K1</td>
<td>XFEM</td>
</tr>
<tr>
<td>K2</td>
<td></td>
</tr>
<tr>
<td>K3</td>
<td></td>
</tr>
<tr>
<td>LAG2_C</td>
<td>XFEM (multi-cracking)</td>
</tr>
<tr>
<td>LAG3_C</td>
<td></td>
</tr>
<tr>
<td>LAG4_C</td>
<td></td>
</tr>
<tr>
<td>LAG2_F1</td>
<td>XFEM (multi-cracking)</td>
</tr>
<tr>
<td>LAG2_F2</td>
<td></td>
</tr>
<tr>
<td>LAG3_F1</td>
<td></td>
</tr>
<tr>
<td>LAG3_F2</td>
<td></td>
</tr>
<tr>
<td>LAG4_F1</td>
<td></td>
</tr>
<tr>
<td>LAG4_F2</td>
<td></td>
</tr>
<tr>
<td>PRE_FLU</td>
<td>HM-XFEM</td>
</tr>
<tr>
<td>PR2_FLU</td>
<td></td>
</tr>
<tr>
<td>PR3_FLU</td>
<td></td>
</tr>
<tr>
<td>LAG_FLI</td>
<td></td>
</tr>
<tr>
<td>LAG_FLS</td>
<td></td>
</tr>
<tr>
<td>LA2_FLI</td>
<td></td>
</tr>
<tr>
<td>LA2_FLI</td>
<td></td>
</tr>
<tr>
<td>LA2_FLI</td>
<td></td>
</tr>
</tbody>
</table>

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 2020 EDF R&D - Licensed under the terms of the GNU FDL (http://www.gnu.org/copyleft/fdl.html)
5.2 Conflicts between the degrees of freedom

Within same **AFFE_CHAR_MECA**, including between the various occurrences, one checks that there is not redundancy between the limiting conditions by application of the rules of overload and remanence, (see §3.8). Indeed, such a situation would lead to a singular matrix during calculation. However, if the same boundary condition is specified twice by two calls **different** with **AFFE_CHAR_MECA** (for example, with two values of imposed displacement), that led to a singular matrix. The limiting conditions kinematics are always imposed on **nodes**, not on the meshes.

5.3 Operations of pairing

5.3.1 Pairing node with node (compatible grids)

This kind of pairing (used for example in **LIAISON_GROUP** or **LIAISON_COQUE**) allows to establish couples of nodes two-with-two. It is in the same way made that in **AFFE_CHAR_THER**. Initially, the two lists of nodes are drawn up \( \Gamma_1 \) and \( \Gamma_2 \) to put in opposite (IE to be paired), for each occurrence of the keyword factor. 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 \( N3 \)), the following error message is transmitted:

  There is a conflict in with respect to the nodes.

  The node \( N1 \) is at the same time it with respect to the node \( N2 \) and of the node \( N3 \).

- 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 \( N3 \)), the following error message is transmitted:

  There is a conflict in with respect to the nodes.

  The node \( N1 \) is at the same time it with respect to the node \( N2 \) and of the node \( N3 \).

- 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 emitted:

  There is a conflict in with respect to the nodes.

  The node \( N1 \) is not the image of any node by the 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 \( \Gamma_1 \) and \( \Gamma_2 \) 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 before calculating the distances (keywords **TRAN**, **CENTER** and **ANGL_NAUT**).

In the couples of nodes in opposite, the order of the nodes is important. So for the first occurrence of the keyword factor, a node \( N \) belonged to the first group of nodes and a node \( M \) with the second group of node, and that for the second occurrence of the same keyword factor, 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.

5.3.2 Pairing mesh-with-node (incompatible grids)

In the continuation of this paragraph, one will speak about the face “slave” (FACE2) and of the face “Master” (FACE1). The “sticking together” of two faces will be done by writing of linear relations between the degrees of freedom of the two faces. Displacements of the nodes of the face slave will be connected to displacements of their projections on the face Master. For each node of the face slave, one will write 2 (in 2D) or 3 (in 3D) linear relations (see operator PROJ_CHAMP for more details).

The principle of the connection is to eliminate the degrees of freedom slaves by writing them like linear relations from the main degrees of freedom. There is a certain symmetry in the problem and one could believe that one can choose randomly who will be the Master and who will be the slave.

Actually, it is necessary to be attentive on two particular points:

- Syntax is not symmetrical: side slave, the user must specify the nodes “to be welded”, whereas main side, it must give meshes. Moreover, the meshes Masters are (for the moment) of a topological dimension with what would be natural. For example, for a grid 2D, surfaces to be restuck are lines, and one could expect that the meshes Masters are segments. The code expects surface meshes (quadrangles and triangles).
- It is preferable (from a mechanical point of view) to choose like surface slave surface with a grid most finely. In the same way that when two sheets are welded, it is to better multiply the points of welding.

**Notice S :**

- In 3D, one should not give meshes Masters of surface, but the meshes voluminal adjacent with the face. The specified meshes are “candidates” for the research of the points opposite. One can give too much of it, that is not awkward. In the same way, in 2D, the meshes “Masters” must be surface (QUAD, SORTED) and nonlinear.
- When one resticks a formed grid by linear elements (P1) on another quadratic grid (P2), it is rather advised to choose like face “slave” the quadratic face.

5.4 Keyword DDL_IMPO

DDL_IMPO= _F ( ✩ /TOUT = ‘YES’, /GROUP_NO = lgrno, /GROUP_MA = lgma, ✗ SANS_GROUP_MA = lgmal, ✗ SANS_GROUP_NO = lgrno1, ✩ /\|DX\| = ux , /LIAISON = ‘EMBEDS’ )

The keyword DDL_IMPO is usable to impose on nodes one or more values of degree of freedom.

- Topological assignment: ALL, GROUP_MY, GROUP_NO, SANS_GROUP_MY, SANS_GROUP_NO

The conditions kinematics are imposed on the nodes given by the keywords ALL, GROUP_MY, GROUP_NO while possibly excluding thanks to the keywords SANS_.

- Components:

  - For AFFE_CHAR_MECA : DX, DY, DZ, DRX, DRY, MARTINI, DRZ, GRX, NEAR, PHI, TEMP, PRE1, PRE2, UI2, UI3, UI4, UI5, UI6, UO2, UO3, UO4, UO5, UO6, VI2, VI3, VI4, VI5, VI6, V2O2, V2O3, V2O4, V2O5, V2O6, WI2, WI3, WI4, WI5, WI6, W2O2, W2O3, W2O4, W2O5, W2O6, W1O1, W1O2, GONF, H1X, H1Y, H1Z, H1PRE1, H2X, H2Y, H2Z, H2PRE1, H3X, H3Y, H3Z, H3PRE1, H4X, H4Y, H4Z, K1, K2, K3, PRE_FLU, LAG_FLI, LAG_FLS, PR2_FLU, LA2_FLI, LA2_FLS, PR3_FLU, LA3_FLI, LA3_FLS, LAGS_C, LAGS_F1, LAGS_F2, LAG2_C, LAG2_F1, LAG2_F2, LAG3_C, LAG3_F1, LAG3_F2, LAG4_C, LAG4_F1, LAG4_F2, V11, V12, V13, V21, V22, V23, V31, V32, V33, PRES11, PRES12, PRES13, PRES21, PRES22, PRES23, PRES31, PRES32, PRES33, LH1 and GL1S.
• For **AFFE_CHAR_MECA_C**: DX, DY, DZ, DRY MARTINI, DRZ, GRX, NEAR, PHI and GLIS.
• DX, DY, DZ, DRX, DRY MARTINI, DRZ, GRX, NEAR, PHI, TEMP, PRE1, PRE2, GONF, H1X, H1Y, H1Z, H1PRE1, K1, K2, K3, LAGS_C and GLIS.

The significance of all these degrees of freedom is specified in the § 5.1.

Note:
• During a calculation with the method X-FEM, it is possible to impose the displacement of nodes nouveau riches. ( **AFFE_CHAR_MECA** only). That is done in a usual way (although these nodes do not have a degree of freedom DX, DY or DZ). If the required node is on the lips, then one imposes the condition of blocking on the nodes of the upper lips and lower.
• The degree of freedom LH1 (hydraulic multiplier of lagrange for the joined elements of type '_JHMS') allows to neutralize the degrees of freedom at the edge of the joint if the solid mass of supports is purely mechanical.
• Degrees of freedom imposed are defined in the reference mark **total** of definition of the grid.

§ CONNECTION = ‘EMBEDS’

Allows to embed nodes directly, i.e. to force to zero the degrees of freedom of translation and rotation. The other degrees of freedom are not modified.

5.5 **Keyword ARETE_IMPO**

```
ARETE_IMPO = F(   ♦ /TOUT    = 'YES',
                  /GROUP_MA    = lgma,   [l_gr_maille]
                  ◊ SANS_GROUP_MA = lgma1,  [l_gr_maille]
                  ◊ SANS_GROUP_NO = lgno1,  [l_gr_noeud]
                  ♦ /|DX      = ux,    [R]
                  ||DY      = uy,    [R]
                  ||DZ      = uz,    [R]
                  ||NEAR    = p,     [R]
                  ||PHI     = phi,   [R]
                  ||TEMP    = T,     [R]
                  ||PRE1    = pr1,   [R]
                  ||PRE2    = pr2,   [R]
                  /|DTAN    = C,     [R])
```

The keyword **ARETE_IMPO** allows to impose all the nodes of an edge on voluminal elements, one or more values of degree of freedom.

♦ **Topological assignment**: ALL, GROUP_MY, SANS_GROUP_MY, SANS_GROUP_NO

The conditions kinematics are imposed on the nodes belonging to the meshes given by the keywords ALL, GROUP_MY while possibly excluding thanks to the keywords SANS_* . The meshes are necessarily segments .

♦ **Components**: DX, DY, DZ, GRX, PRE 1, PRE 2, NEAR, PHI, TEMP

The significance of these degrees of freedom is specified in the § 5.1.

Note:
• Degrees of freedom imposed are defined in the reference mark **total** of definition of the grid. If one wants to impose a degree of freedom in another direction, it is possible to use the keyword DTAN.

♦ **DTAN**

Allows to apply a limiting condition in the tangent direction to the edge (see § 3.9.1). One thus modifies the values of the degrees of freedom of displacement DX, DY and DZ.

**Example**:

```
ARETE_IMPO = ( _F(GROUP_NO = 'LowSide',
                 DX = 0, DY = 0, DZ = 0),
               _F(GROUP_MA = 'RightSide', SANS_GROUP_NO = 'Corner'),
```
Significance of the second occurrence of ARETE_IMPO is: “for all the nodes of the group of meshes ‘RightSide’, DTAN = 10 except for those of the group of nodes ‘Corner’. This makes it possible not to have boundary conditions redundant.

5.6 Keyword FACE_IMPO

The keyword FACE_IMPO allows to impose on all the nodes of a face on surface or voluminal elements, one or more values of degree of freedom.

♦ Topological assignment: ALL, GROUP_MY, SANS_GROUP_MY, SANS_GROUP_NO
The conditions kinematics are imposed on the nodes belonging to the meshes given by the keywords ALL, GROUP_MY while possibly excluding thanks to the keywords SANS_*. The meshes are necessarily triangles or of the quadrangles in 3D and the segments in 2D.

♦ Components: DX, DY, DZ, DRX, DRY MARTINI, DRZ, GRX, PRE1, PRE2, NEAR, PHI, TEMP
The significance of these degrees of freedom is specified in the § 5.1.

Note:
• Degrees of freedom imposed are defined in the reference mark total of definition of the grid.
  If one wants to impose a degree of freedom in another direction, it is possible to use the keyword DTAN.

♦ DTAN
Allows to apply a limiting condition in the tangent direction to the face (see § 3.9.1). One thus modifies the values of the degrees of freedom of displacement DX, DY and DZ. This condition can function only in 2D, not in 3D.

♦ DNOR
Allows to apply a limiting condition in the normal direction with the face (see § 3.9.1). One thus modifies the values of the degrees of freedom of displacement DX, DY and DZ.

Notice concerning the keywords DNOR and DTAN:
If the geometry of the meshes (of skin) selected is not good quality, the calculation of the normals can be disturbed and that can lead to "kicked up a rumpus" results. If the zone concerned is plane, one can then replace the keyword FACE_IMPO/DNOR (or DTAN) by the word key LIAISON_OBLIQUE. One is then sure that the nodes will move rigorously in the same direction.
5.7 Keyword LIAISON_DDL

```
LIAISON_DDL =_F(    ♦ GROUP_NO    =  lgno,   [l_gr_noeud]
                ♦ DDL    =  lddl,   [l_K8]

If AFFE_CHAR_MECA
  ♦ COEF_MULT    =  has,,   [l_R]
  ♦ COEF_IMPO    =  B,,  [R]

If AFFE_CHAR_MECA_C
  ♦ COEF_MULT    =  has,,   [l_R]
  ♦ COEF_IMPO    =  B,,  [C]

If AFFE_CHAR_MECA_F
  ♦ /  COEF_MULT_ FONC  =  has ,, [ function ]
  ♦ /  COEF_IMPO     =  B,,  [ function ]
)
```

The keyword LIAISON_DDL allows to define a linear relation between degrees of freedom of two or several nodes. The following kinematic condition will be applied:

\[
\sum_{i=1}^{r} \alpha_i U_i = \beta
\]

With \( U_i \) the list of \( r \) Degrés de freedom, \( \alpha_i \) coefficients and \( \beta \) the second member.

- **Topological assignment: GROUP_NO**
  Give the list of the nodes \( N_i \) ordinate in a natural way:
  - In the order of the list of groups of nodes, and for each group of nodes, in the order of definition of the group by GROUP_NO,
  
  Caution! The order of the nodes has an importance (see the examples).

- **DDL**
  Give the list ordinate \( r \) Degrés de freedom \( U_i \) (see §5.1 for the possible degrees of freedom).
  Caution! The order of the degrees of freedom has an importance (see the examples).

- **COEF_MULT**
  Give the list ordinate \( r \) coefficients realities \( \alpha_i \).
  Caution! The order of the coefficients has an importance (see the examples).

- **COEF_MULT_FONC**
  Give the list ordinate \( r \) coefficients functions \( \alpha_i \) (AFFE_CHAR_MECA_F). The functions can depend that initial geometry (see § 3.2.1).
  Caution! The order of the coefficients has an importance (see the examples).

- **COEF_IMPO = has :**
  Value of the linear relation \( \beta \). If it is a function (AFFE_CHAR_MECA_F), this one can depend on time or of the initial geometry (see § 3.2.1).

**Example 1**: one wants to impose one linear relation between the degrees of freedom of one even node

In this case, typical case one will repeat behind the keyword GROUP_NO the name (of group) of node as many times as there are degrees of freedom in the relation. Example: to impose \( U_x = U_y \) on the node of name (of group) \( GN1 \), one will write:

```
LIAISON_DDL =_F( GROUP_NO = ('GN1', 'GN1'),
  DDL = ('DX', 'DY'),
  COEF_MULT = (1., -1.),
  COEF_IMPO = 0.,)
```
Example 2: one wants to impose a linear relation between groups of nodes. 

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 two groups of nodes `GRN01` and `GRN02` (even displacement \( U \), node with node for example) one cannot write:

```plaintext
LIAISON_DDL = _F ( GROUP_NO = ('GRN01', 'GRN02'), 
DRL = ('DX', 'DX'), 
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 above to clarify each linear relation, node by node, or to use `LIAISONGROUP` [§ 5.12] which makes it possible to condense the writing of same linear relations between two groups of nodes out of screw - with - screw.

5.8 **Keyword LIAISON_OBLIQUE**

```plaintext
LIAISON_OBLIQUE = _F ( 
  ▲ GROUP_NO = lgNo, [l_gr_noeud] 
  ▲ | DX = ux, [R] or [function] 
  ▲ | DY = uy, [R] or [function] 
  ▲ | DZ = uz, [R] or [function] 
  ▲ | DRX = RX, [R] or [function] 
  ▲ | DRY MARTINI = R there, [R] 
  ▲ or [function] 
  ▲ | DRZ = R Z, [R] or [function] 
  ▲ [function] 
  ▲ ANGL_NAUT = (has, B, G), [l_R])
```

The keyword `LIAISON_OBLIQUE` allows to apply, with nodes or groups of nodes, the same component value of displacement definite per component in an unspecified oblique reference mark.

- **Topological assignment:** `GROUP_NO`
The loading is affected on the nodes.

- **Components:** \( dx, Dy, dz, X-ray, ry, rz \) 
  Values of the components.

- **ANGL_NAUT** 
  List of the three angles, in degrees, which define the oblique reference mark of application of the degrees of freedom (the last angles of the list can be omitted if they are worthless). The nautical angles make it possible to pass from the total reference mark of definition of the coordinates of the grid to an unspecified oblique reference mark (see §3.9.3). By defaults the angles are identically worthless and thus the components of forces are defined in the reference mark total.

5.9 **Keyword LIAISON_UNIF**

```plaintext
LIAISON_UNIF = _F ( 
  ▲ /GROUP_NO = lgno, [l_gr_noeud] 
  ▲ /GROUP_MA = lgma, [l_gr_maille] 
  ▲ ◊ SANS_GROUP_MA = lgmal, [l_gr_maille] 
  ▲ ◊ SANS_GROUP_NO = lgnol, [l_gr_noeud] 
  ▲ ◊ DDL = lddl, [l_K8] )
```

The keyword `LIAISON_UNIF` allows to impose with all the degrees of freedom of all the provided nodes the same value (unknown). One will have thus:
$U_i(N_k) = U_i(N_j)$ for all the degrees of freedom $U_i$ indicated on the nodes $N_k$ (3)

♦ Topological assignment: GROUP_MA, GROUP_NO, SANS_GROUP_MA, SANS_GROUP_NO

The conditions kinematics are imposed on the nodes $N_k$ defined by the keyword GROUP_NO or those belonging to the meshes given by word-Clé GROUP_MA, without the nodes defined under SANS_GROUP_.*

♦ DDL

Give list of Degrees of freedom $U_i$ (see §5.1 for the possible degrees of freedom).

5.10 Keyword LIAISON_CHAMNO

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

The keyword LIAISON_CHAMNO allows to define a linear relation between all the degrees of freedom present in a concept CHAM_NO. This keyword can be also used to impose on the structure (or a part) a given work, for a loading calculated as a preliminary with another AFFE_CHAR_MECA and leading to an assembled vector produces by ASSE_VECTEUR [U4.61.23].

♦ CHAM_NO
Name of cham_no who is used to define the linear relation. The degrees of freedom connected are all those present in cham_no. The coefficients to be applied to the degrees of freedom are the values of cham_no for these degrees of freedom.

♦ COEF_IMPO
Value of the real coefficient $\beta$ applied to the second member of the linear relation.

◊ NUME_LAGR
See § 4.4.

Example:

Let us suppose that one has one cham_no relating to two nodes of name $N01$ and $N02$ respectively carrying the degrees of freedom ‘DX’, ‘DY’, and ‘DZ’ for the $N01$ node and ‘DX’, ‘DY’, ‘DZ’, ‘DRX’, ‘DRY MARTINI’ and ‘DRZ’ for the node $N02$.

Also let us suppose that it cham_no has the following values for these degrees of freedom:
‘DX’ $N01$ 2.
‘DY’ $N01$ 1.
‘DZ’ $N01$ 3.
‘DX’ $N02$ 1.
‘DY’ $N02$ 4.
‘DZ’ $N02$ 2.
‘DRX’ $N02$ 3.
‘DRY MARTINI’ $N02$ 5.
‘DRZ’ $N02$ 2.

The linear relation that one will impose is:

\[
2.\cdot{\text{DX}}(N01) + 1.\cdot{\text{DY}}(N01) + 3.\cdot{\text{DZ}}(N01) + 1.\cdot{\text{DX}}(N02) + 4.\cdot{\text{DY}}(N02) + 2.\cdot{\text{DZ}}(N02) + 3.\cdot{\text{DRX}}(N02) + 5.\cdot{\text{DRY}}(N02) + 2.\cdot{\text{DRZ}}(N02) = B
\]
5.11 Keyword CHAMNO_IMPO

CHAMNO_IMPO = _F (
   ♦ CHAM_NO = chamno, [cham_no]
   ♦ COEF_IMPO = B, [R]
   ◊ NUME_LAGR = / ‘NORMAL’, [DEFECT]
                  / ‘AFTER’)

It is by way of a light adaptation the keyword LIAISON_CHAMNO (see §5.10). This one makes it possible to apply like coefficients of linear relation the contents of one cham_no. In the case of the keyword CHAMNO_IMPO, one takes the contents of one cham_no like second member of the linear relation. It is thus strictly equivalent to a manual procedure where one recovers the values of cham_no on the hand then one imposes them via DDL_IMPO. If the field contains value 2 for DX, the relation is imposed: \( DX = COEF\_IMPO \times 2 \). The coefficient \( \alpha \) is fixed at 1.

♦ CHAM_NO
   Name of cham_no who is used to define the second members of the linear relation.

♦ COEF_IMPO = B
   Multiplying coefficient of cham_no.

◊ NUME_LAGR
   See § 4.4.

5.12 Keyword LIAISON_GROUP

LIAISON_GROUP = _F (  
   ♦ / GROUP_MA_1 = lgma1, [l_gr_maille]
   GROUP_MA_2 = lgma2, [l_gr_maille]
   / GROUP_NO_1 = lgn01, [l_gr_noeud]
   GROUP_NO_2 = lgn02, [l_gr_noeud]
   ◊ SANS_GROUP_NO = lgn0, [l_gr_noeud]
   ♦ DDL_1 = /| ‘DX’,
           | ‘DY’,
           | ‘DZ’,
           | ‘DRX’,
           | ‘DRY MARTINI’,
           | ‘DRZ’,
           /| ‘DNOR’,
   ♦ DDL_2 = /| ‘DX’,
           | ‘DY’,
           | ‘DZ’,
           | ‘DRX’,
           | ‘DRY MARTINI’,
           | ‘DRZ’,
           /| ‘DNOR’,
   ♦ COEF_MULT_1 = has \text{ii}, [l_R]
   ♦ COEF_MULT_2 = has \text{ii}, [l_R]
   ♦ COEF_IMPO = B, [R] or [function]
   ◊ TOP = ‘YES’,
   ◊ CENTER = center, [l_R]
   ◊ ANGL_NAUT = (has, B, G), [l_R]
   ◊ TRAN = (X, there, Z), [l_R])

The keyword LIAISON_GROUP allows to define the same linear relation between certain degrees of freedom of couples of nodes, these couples of nodes being obtained while putting in opposite two lists of meshes or nodes (see §5.3.1). The full number of imposed relations is equal to the number of couple of nodes.

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 2020 EDF R&D - Licensed under the terms of the GNU FDL (http://www.gnu.org/copyleft/fdl.html)
Topological assignment

GROUP_MA_1, GROUP_NO_1:
First list of nodes to be connected (noted $\Gamma_1$).

GROUP_MA_2, GROUP_NO_2:
Second list of nodes to be connected (noted $\Gamma_2$).

SANS_GROUP_NO:
Itt operand makes it possible to remove list of the couples of nodes out of screw - with screw, i.e. all the couples of which at least one of the nodes belongs to the list of nodes described by itt operand. That makes it possible to avoid the accumulation of linear relations on the same node with cours de various repetitions of the keyword factor LIAISON_GROUP, what leads most of the time to a singular matrix.

DDL_1
Give list of Degrés of freedom for the edge $\Gamma_1$ (see § 5.1 for the possible degrees of freedom).
If DDL_1 = 'DNOR', one binds the degrees of freedom of displacement according to the normal to the surface of the element (see § 3.9.1).

DDL_2
Give the list of the degrees of freedom for the edge $\Gamma_2$ (see § 5.1 for the possible degrees of freedom). If DDL_2 = ‘DNOR’, one binds the degrees of freedom of displacement according to the normal to the surface of the element (see § 3.9.1).

COEF_MULT_1
List of realities dimensioned exactly with the number of degrees of freedom declared in DDL_1 corresponding to the multiplying coefficients of the linear relation.

COEF_MULT_2
List of realities dimensioned exactly with the number of degrees of freedom declared in DDL_2 corresponding to the multiplying coefficients of the linear relation.

COEF_IMPO
Coefficient of blocking of the linear relation.

CENTRE/ANGL_NAUT/TRAN
Operands CENTER / ANGL_NAUT / TRAN (see §4.6) allow to define a virtual transformation (rotation and/or translation) approximate of $\Gamma_1$ in $\Gamma_2$ in order to ensure the bijectivity of the function opposite [§5.3.1]. The order carries out initially rotation, then the translation.

TOP = ‘YES’
When the meshes of edge are quadratic (thus SEG3) the use of SOMMET=' OUI' force the algorithm of pairing to associate the tops of SEG3 at other tops, and the mediums of SEG3 in other mediums. In the case of fine grids, that makes it possible in certain cases to avoid the problems of conflicts of opposite.

Example:
One wants to impose a cyclic condition of repetitivity (even normal displacement) between the face 1 and the face 2 geometry below:
Let us suppose that $\text{FACE1}$ (respectively $\text{FACE2}$) that is to say made up of the list of meshes $lma1$ (respectively $lma2$). As the relation must be bijective, the two faces comprise necessarily the same number of nodes $nbno$. One wants to write the following linear relations:

$$\forall \, N_i^1 \, \text{node of the face 1 of opposite} \, N_i^2$$

$$\forall \, i=1,...,nbno$$

$$u.n\big{|N_i^1\big|} = u.n\big{|N_i^2\big|}$$  \hfill (4)

Data of $\text{LIAISON_GROUP}$ will be written:

```plaintext```
LIAISON_GROUP= _F ( GROUP_MA_1 = lma1,
GROUP_MA_2 = lma2,
DDL_1 = 'DNOR',
DDL_2 = 'DNOR',
COEF_MULT_1 = 1.,
COEF_MULT_2 = -1.,
COEF_IMPO = 0,
CENTER = (X0, Y0, Z0),
ANGL_NAUT = (\alpha, 0., 0.),
)
```

In the case or $\text{FACE2}$ is perpendicular to axis X, the preceding example can be also written:

```plaintext```
LIAISON_GROUP= _F ( GROUP_MA_1 = lma1,
GROUP_MA_2 = lma2,
DDL_1 = ('DX', 'DY'),
DDL_2 = 'DY',
COEF_MULT_1 = (- \sin (\alpha), \cos (\alpha)),
COEF_MULT_2 = -1.,
COEF_IMPO = 0
)
```

Note:

The full number of relations imposed by an occurrence does not depend amongst arguments on $\text{DDL}_1$ or $\text{DDL}_2$, which are only used to enrich the relation.

In the following case, one imposes $nbno$ relations:

```plaintext```
LIAISON_GROUP= _F ( GROUP_MA_1 = lma1,
GROUP_MA_2 = lma2,
DDL_1 = ('DX', 'DY'),
DDL_2 = ('DX', 'DY'),
COEF_MULT_1 = (1, 1),
COEF_MULT_2 = (-1, -1),
COEF_IMPO = 0
)
```

Dyears the following case, one impose $2\times nbno$ relations:

```plaintext```
LIAISON_GROUP= _F ( GROUP_MA_1 = lma1,
GROUP_MA_2 = lma2,
DDL_1 = 'DX',
DDL_2 = 'DX',
COEF_MULT_1 = 1,
```

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 2020 EDF R&D - Licensed under the terms of the GNU FDL (http://www.gnu.org/copyleft/fdl.html)
COEF_MULT_2 = -1.,
COEF_IMPO = 0)

\_F ( 
  GROUP_MA_1 = lma1,
  GROUP_MA_2 = lma2,
  DDL_1 = 'DY',
  DDL_2 = 'DY',
  COEF_MULT_1 = 1,
  COEF_MULT_2 = -1.,
  COEF_IMPO = 0
)

It should well be noted that these the last two examples are not equivalent.
5.13 **Keyword LIAISON_MAIL**

```
LIAISON_MAIL = F(
  ♦ | GROUP_MA_ESCL = lgma1, [l_gr_maille]
  ♦ | GROUP_NO_ESCL = lgnol, [l_gr_noeud]
  ♦ GROUP_MA_MAIT = lgma1, [l_gr_maille]
  ◊ TYPE_RACCORD = / 'MASSIVE' [DEFECT]
                  / 'HULL'
                  / 'COUE MASSIF'
                  / 'MASSIF_COQUE'
  ◊ ELIM_MULT = / 'NOT', [DEFECT]
             / 'YES',
  ◊ DISTANCE_MAX = d_max, [R]
  ◊ DISTANCE_ALARME = d_ala, [R]

# if TYPE_RACCORD = 'MASSIVE'
  ◊ ANGL_NAUT = (has, B, c) [l_R]
  ◊ CENTER = (cx, cy, cz) [l_R]
  ◊ TRAN = (tx, ty, tz) [l_R]
  ◊ DDL_MAIT = 'DNOR',
  ◊ DDL_ESCL = 'DNOR',

# if TYPE_RACCORD = 'COUE_MASSIF'
  ♦ THICK = thick, [l_R]
  ♦ CHAM_NORMALE = chanor, [cham_no])
```

5.13.1 **Use and examples**

Notice (cf § 5.3.2) :

_In 3D, one should not give meshes Masters of surface, but them voluminal meshes adjacent with the face. The specified meshes are “candidates” for the research of the points opposite. One can give too much of it, that is not awkward. In the same way, in 2D, the meshes “Masters” must be surface (QUAD, SORTED) and nonlinear._

The keyword LIAISON_MAIL allows to define linear relations “to restick” two “edges” of a structure. The characteristic of this keyword (compared to LIAISON_GROUP for example) is to allow to bind displacements of unconstrained nodes on the grid. Grids of FACE1 and FACE2 can be incompatible.

**Examples** :

a) a condition of periodicity (study of a cell of homogenisation)  

```
FACE 1

      /
      |
      /
FACE 2
```

The experiment showed that for calculations of periodic homogenisation, the results are much more precise if the 2 faces have compatible grids (i.e. that grids of FACE1 and FACE2 are superposable modulos a isometry).
b) a cyclic condition of repetitivity

\begin{center}
\begin{tikzpicture}
\draw (0,0) -- (2,1);
\draw (0,0) -- (1,2);
\draw (1,2) -- (2,1);
\draw (3,0) -- (2,1);
\draw (3,0) -- (4,1);
\end{tikzpicture}
\end{center}

\begin{center}
\begin{tikzpicture}
\draw (0,0) -- (2,1);
\draw (0,0) -- (1,2);
\draw (1,2) -- (2,1);
\end{tikzpicture}
\end{center}

\begin{center}
\begin{tikzpicture}
\draw (0,0) -- (2,1);
\draw (0,0) -- (1,2);
\draw (1,2) -- (2,1);
\draw (3,0) -- (2,1);
\draw (3,0) -- (4,1);
\end{tikzpicture}
\end{center}

\begin{center}
\begin{tikzpicture}
\draw (0,0) -- (2,1);
\draw (0,0) -- (1,2);
\draw (1,2) -- (2,1);
\end{tikzpicture}
\end{center}

c) a condition of simple sticking together

\begin{center}
\begin{tikzpicture}
\draw (0,0) -- (2,1);
\draw (0,0) -- (1,2);
\draw (1,2) -- (2,1);
\draw (3,0) -- (2,1);
\draw (3,0) -- (4,1);
\end{tikzpicture}
\end{center}

In the continuation of this paragraph, one will speak about the face “slave” (FACE2) and of the face “Master” (FACE1). The “sticking together” of the two faces will be done by writing of linear relations between the degrees of freedom of the two faces, to see §5.3.2.

Displacements of the nodes of the face slave will be connected to displacements of their projections on the face Master. For each node of the face slave, one will write two (in 2D) or three (in 3D) linear relations.

5.13.2 Some remarks and precautions of use

- If FACE1 and FACE2 are not geometrically confused but that there exists a isometry (rotation and translation) between the two, the user must define this isometry (that which transforms FACE2 in FACE1), thanks to the keywords CENTER/ANGL_NAUT/TRAN.

- A “classical” use of this functionality is for example the sticking together of a formed grid by linear elements (P1) on another quadratic grid (P2). In this case it is rather advised to choose like face “slave” the quadratic face.

- Keyword LIAISON_MAIL is made in theory to connect two surfaces a priori disjoined. Sometimes it is not the case and a node slave can belong to the one of the meshes Masters. The linear relation that the problem seeks to write becomes a tautology (X = X), which leads to a null pivot during factorization. To avoid this problem, one does not write the relations connecting a node slave to his mesh Master if:
  - This node belongs to the connectivity of the mesh;
  - And if keywords CENTER/ANGL_NAUT/TRAN were not used.

- It is necessary to be conscious that for each occurrence of LIAISON_MAIL, one connects a priori all Lbe nodes main slaves with the meshes even if the distances from projection are important (one emits however alarms in this case). One can change this behavior into using the keyword DISTANCE_MAX.

- It is not possible to use the connection MASSIF_COQUE with elements of the type COQUE_3D for the hull part.

- If one writes:
  \begin{verbatim}
  LIAISON_MAIL = ( _F (GROUP_MA_ESCL=' GE', GROUP_MA_MAIT = ' GM1'),
                 _F (GROUP_MA_ESCL=' GE', GROUP_MA_MAIT = ' GM2'))
  \end{verbatim}

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 2020 EDF R&D - Licensed under the terms of the GNU FDL (http://www.gnu.org/copyleft/fdl.html)
It would be an error to think that the program will sort in GE nodes close to GM1 and those close to GM2. In this example, nodes of GE will be eliminated twice and one can expect a problem of null pivot during factorization.

The user must write:

\[
\text{LIAISON_MAIL = } _F (\text{GROUP_MA_ESCL='GE', GROUP_MA_MAIT= ('GM1', 'GM2'))}
\]

5.13.3 Keywords

♦ Topological assignment

\[
\text{GROUP_MA_ESCL, GROUP_NO_ESCL :}
\]

List of nodes slaves to connect. When one wants to restick only normal displacements of the faces (cf keywords DDL_MAIT and DDL_ESCL), it is necessary to be able to determine the normal direction of the faces. The normal direction is calculated on the face slave. It is thus necessary in this case to use the keyword GROUP_MA_ESCL with meshes of the type “facets”.

\[
\text{GROUP_MA_MAIT :}
\]

List of meshes Masters to connect.

◊ DISTANCE_MAX

When that a node slave is not (geometrically) in any mesh Master, the program connects the node and the point (of the edge) of the mesh nearest.

If it is wished that a node “remote” slave of the meshes Masters not be concerned with the connection, the operand should be used DISTANCE_MAX. This operand makes it possible to give the maximum distance beyond which one will not make a connection.

There is no value by default for DISTANCE_MAX. What wants to say that by default, the connection will relate to all the nodes slaves.

◊ DISTANCE_ALARME = d_ala

The keyword DISTANCE_ALARME = d_ala allows to be alarmed if a node slave is at a distance \(D > d_ala\) mesh Master nearest.

There is no value by default for DISTANCE_ALARME. By default, one emits an alarm if the distance \(D\) than 1/10ème nearest size of the mesh is higher (relative criterion).

◊ TYPE_RACCORD

This keyword leaveand of to choose the type of the linear relations which one will write to eliminate the degrees of freedom from the nodes slaves.

• If TYPE_RACCORD='MASSIF', the nodes are supposed to carry degrees of freedom of translation (DX, DY, DZ). If the user does not specify DDL_MAIT='DNOR', one will write (for example in 2D), two linear relations for each node slave: one to eliminate sound ‘DX’, the other to eliminate sound ‘DY’.

• If TYPE_RACCORD='COQUE', the nodes are supposed to carry degrees of freedom of translation (DX, DY, DZ) and of the degrees of freedom of rotation (DRX, DRY MARTINI, DRZ). Six will be written linear relations to eliminate them six degrees of freedom of each node slave.

• If TYPE_RACCORD='MASSIF_COQUE', the nodes slaves are supposed “massive” (translations: DX, DY, DZ) and nodes Masters are supposed of standard “hull” (three translations and three rotations). Lbe degrees of freedom of translation of the nodes slaves are eliminated by writing that they are equal to the translations of the point “Master” in opposite. Translations of the point Master are calculated as if the small segment of normal to the hull remained rigid.

• If TYPE_RACCORD='COQUE_MASSIF', the nodes slaves are supposed of standard “hull” (six degrees of freedom: DX, DY, DZ, DRX, DRY MARTINI, DRZ) and nodes Masters are supposed of “massive” type (DX, DY, DZ). The degrees of freedom of translation of the
nodes slaves are eliminated by writing that they are equal to the translations of the point "Master" in opposite. The degrees of freedom of rotation of the nodes slaves are eliminated by writing that they are equal to rotations of the point "Master" in opposite (A). Rotations of the point A are calculated starting from the translations of two other points A1 and A2 located at +h/2 and −h/2, if h is a normal vector with the hull and of which the length is the thickness of the hull (see keywords THICK and CHAM_NORMALE).

◊ DDL_MAIT/ DDL_ESCL

If one wants to restick only normal displacements with the faces, it is necessary to specify DDL_MAIT= 'DNOR' and DDL_ESCL=' DNOR'. The normal direction being calculated on the face slave, it is necessary to give meshes of facet, to see GROUP_MY_ESCL). This normal direction is transformed by the possible rotation of the geometrical transformation (see CENTRE/ANGL_NAUT/TRAN) to determine the normal direction on the face Master.

◊ CENTRE/ANGL_NAUT/TRAN

Operands CENTER / ANGL_NAUT / TRAN (see § 4.6 ) allowing to pass from the face main slave to the face . The order carries out initially rotation, then the translation. If these keywords are absent, it is that the geometrical transformation is "the identity" i.e. the faces Master and slave are geometrically confused. Caution! The direction of the transformation is slave towards Master .

◊ THICK/CHAM_NORMALE

These two keywords are obligatory if TYPE_RACCORD = ‘COQUE_MASSIF ’:THICK give the thickness of the hull on the level of the connection (presumably constant) and CHAM_NORMALE give a field with the nodes which contains the direction of the normal to the hull on the nodes of the meshes " slaves ". The field can be obtained by the order:

CHNOR = CREA_CHAMP ( TYPE_CHAM = ‘NOEU_GEOM_R’, OPERATION = ‘NORMAL’, MODEL = MODEL, GROUP_MA = ‘GMCOQU’)

◊ ELIM_MULT

This keyword is used to solve the problem which can be posed when several surfaces adjacent slaves are restuck (i.e. which have one or more common nodes). Let us imagine for example that one writes (in 2D):

LIAISON_MAIL= ( _F (GROUP_MA_ESCL=' LIGNE_AB', GROUP_MAIT=...) _F (GROUP_MA_ESCL=' LIGNE_BC', GROUP_MAIT=...) )

If the user forces ELIM_MULT=' OUI', the program will treat each occurrence of LIAISON_MAIL in way independent. The node B, pertaining to LIGNE_AB and LIGNE_BC will be eliminated twice and it is unfortunately probable that calculation will stop during the factorization of the matrix with the message "Pivot almost no one... " because linear relations generated by LIAISON MAILLE are redundant. Most of the time, the defect (ELIM_MULT=' NON') is the good choice. The only case where the user could use ELIM_MULT=' OUI' is that of the use of the keyword DDL_ESCL=' DNOR' because so in the two occurrences, normal "the slaves" are not the same ones, elimination is not redundant.

5.14 Keyword LIAISON_PROJ

5.14.1 Goal

Keyword factor which makes it possible to define linear relations between the nodes of the same model. The coefficients of the linear relations are given using the order PROJ_CHAMP who allows to have like concept result the matrix of the coefficients of influence determined starting from the
functions of form of the elements. It is the same algorithm which is used with the key word LIAISON_MAIL where one finds the concept of mesh Master and slave whom one also finds in the order PROJ_CHAMP.

5.14.2 Syntax (AFFE_CHAR_MECA only)

LIAISON_PROJ = _F (  
  ♦ MATR_PROJECTION = chamno, [corresp_2_mailla] 
  ♦ DDL = ['DX','DY','DZ','DRX','DRY','DRZ'] 
  ◊ STANDARD = ['IDENTITY', 'EXCENTREMENT'] [defect] )

5.14.3 Operands

♦ MATR_PROJECTION
Name of the concept resulting from the order PROJ_CHAMP. The concept corresponds to the matrix of the coefficients obtained by the option PROJECTION = ‘NOT’ order PROJ_CHAMP.

Example:

matcoeff = PROJ_CHAMP (  
  PROJECTION = ‘NOT’, METHOD = ‘COLLOCATION’,  
  MAILLAGE_1 = e-mail, MAILLAGE_2 = e-mail,  
  VIS_A_VIS = _F (GROUP_MA_2 = 'ARMAT', GROUP_MA_1 = 'SDALLE'),),  
)

MAILLAGE_1 and MAILLAGE_2 must correspond to the same grid. The order AFFE_CHAR_MECA require to inform the model which is based on a grid. As one will write relations between degrees of freedom of a model, it is obligatory that the grids are identical. Coherence is checked during the use of LIAISON_PROJ.

VIS_A_VIS : under this key word factor the groups make it possible to define the meshes Masters and slaves:

• GROUP_MA_1 corresponds to the meshes Masters.
• GROUP_MA_2 corresponds to the meshes slaves.

♦ DDL
List degrees of freedom of the nodes slaves on which one imposes the relations. The existence of the degrees of freedom on the nodes slaves and Masters is checked.

◊ TYPE = ‘OFFSETTING’
In the case of makes it possible existence of degrees of freedom of rotation on the meshes Masters to impose displacements on the nodes slaves which take account of these rotations.

The relations are following form:

• For the degrees of freedom of the nodes slaves given under DDL.

  \[ DDL(N_{escl}) = \sum_i \text{Coeff}_i \times DDL(N_{maître}^i) \]

  with I: main node of the mesh containing the node slave.

• If TYPE= ’EXCENTREMENT’, the relation on the degrees of freedom of translation of the nodes slaves becomes:

  \[ DDL(N_{escl}) = \sum_i \text{Coeff}_i \times (DDL(N_{maître}^i) + \omega(N_{maître}^i) \wedge N_{maître}^i N_{escl}) \]

The existence of the degrees of freedom on the main nodes and slaves is checked.

5.15 Keyword LIAISON_CYCL

LIAISON_CYCL = _F (  
  ♦ | GROUP_NO_ESCL = lgn2, [lgr_noeud] )
The keyword **LIAISON_CYCL** allows to define the linear relations allowing to impose conditions of cyclic symmetry with taking into account of a dephasing. It is mainly dedicated to being used within the restrictive framework of dynamic calculation with cyclic symmetry. The characteristic of this keyword (to the image of **LIAISON_MAIL**) is to allow to bind displacements of unconstrained nodes on the grid. Grids of **FACEG** and **FACED** can be incompatible.

The cyclic condition of repetitivity applied within the framework of dynamics is based on the method of duplication of grid. Operator thus leaves on the postulate that the initial grid of a sector is duplicated in two grids identical to the image of the following figure.

![Cyclic Symmetry Diagram](image)

In the continuation of this paragraph, one will speak about the face “slave” and the face “Master”. The “sticking together” of the two faces will be done by writing of linear relations between the degrees of freedom of the two faces.

Displacements of the nodes of the face slave will be connected to displacements of their projections on the face Master. For each node of the face slave, one will write two (in 2D) or three (in 3D) linear relations.

If **FACEG** and **FACED** are not geometrically confused but that there exists a isometry (rotation + translation) between the two, the user must define this isometry (that which transforms **FACEG** in **FACED**) thanks to the keywords **CENTRE/ANGL_NAUT/TRAN**.

The expression of the cyclic condition of symmetry for a dephasing AND element \( \beta \) given and while considering \( G \) as the interface slave is the following one:

\[
\begin{bmatrix}
q_1^G \\
q_2^G
\end{bmatrix} =
\begin{bmatrix}
\cos \beta & \sin \beta \\
-\sin \beta & \cos \beta
\end{bmatrix}
\begin{bmatrix}
q_1^D \\
q_2^D
\end{bmatrix}
\]  

(5)

In order to write the linear relations making it possible to take into account this condition, it is necessary to give two occurrences of the keyword factor **LIAISON_CYCL**:

- The first makes it possible to bind the degrees of freedom of the face \( G \) grid 1 with the face \( D \) same grid and the face \( D \) grid 2. Coefficients (\( \cos \beta \) and \( \sin \beta \)) must be well informed by the keywords **COEF_MAIT1**, **COEF_MAIT2**, **COEF_ESCL**.

**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 2020 EDF R&D - Licensed under the terms of the GNU FDL (http://www.gnu.org/copyleft/fdl.html)
• The second makes it possible to bind the degrees of freedom of the face \( G \) grid 2 with the face \( D \) same grid and the face \( D \) grid 1. Coefficients \((-\sin \beta \text{ and } \cos \beta)\) must be well informed by the keywords `COEF_MAIT1, COEF_MAIT2`.

♦ **Topological assignment**

GROUP_NO_ESCL/GROUP_MA_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 keyword `GROUP_NO_ESCL` more all nodes carried by the meshes specified by the keyword `GROUP_MA_ESCL`.

GROUP_MA_MAIT_1 :

This keyword makes it possible to define the whole of the meshes Masters of the grid 1 (or 2) where will be sought they with respect to the nodes of the face slave of grid 1 or 2.

GROUP_MA_MAIT_2 :

This keyword permetde to define the whole of the meshes Masters of the grid 1 (or 2) where will be sought they with respect to the nodes of the face slave of grid 1 or 2.

♦ **COEF_MAIT_1/COEF_MAIT_2/COEF_ESCL**

These keywords make it possible to define the coefficients of the linear relation to apply, in the case of cyclic symmetry it is the cosine and sines the angle dephasing AND element considered. These coefficients must thus be coherent with the definition of the interfaces Masters and slaves. The coefficient `COEF_ESCL` allows to pass a coefficient in front of the degrees of freedom slaves. For example:

\[
COEF\_ESCL\begin{bmatrix} q^1_s \\ q^2_s \end{bmatrix} = \begin{bmatrix} COEF\_MAIT1 \times COEF\_MAIT2 \end{bmatrix} \begin{bmatrix} q^1_d \\ q^2_d \end{bmatrix} = \cos \beta \cdot \sin \beta \begin{bmatrix} q^1_d \\ q^2_d \end{bmatrix}
\]

♦ **DDL_MAIT/DDL_ESCL**

If one wants to restick only normal displacements with the faces, it is necessary to specify `DDL_MAIT= 'DNOR'` and `DDL_ESCL= 'DNOR'`. The normal direction being calculated on the face slave, it is necessary to give meshes of facet, to see `GROUP_MY_ESCL`). This normal direction is transformed by the possible rotation of the geometrical transformation (see `CENTRE/ANGL_NAUT/TRAN`) to determine the normal direction on the face Master.

♦ **CENTRE/ANGL_NAUT/TRAN**

Operands `CENTER` / `ANGL_NAUT` / `TRAN` (see § 4.6 ) allowing to pass from the face main slave to the face . The order carries out initially rotation, then the translation. If these keywords are absent, it is that the geometrical transformation is “the identity” i.e. the faces Master and slave are geometrically confused. Caution! The direction of the transformation is slave towards Master.

### 5.16 Keyword LIAISON_SOLIDE

```
LIAISON_ SOLID = _F (    ♦ / GROUP_MA = lhma, [l_gr_maille]  
/ GROUP_NO = lg No , [l_gr_noeud]  
♦ DIST_MIN = dmin, [R]  
♦ NUME_LAGR = / 'NORMAL', [DEFECT]  
/ 'AFTER' )
```

The keyword `LIAISON_SOLIDE` allows to model an indeformable part of a structure. One imposes linear relations between the degrees of freedom of the nodes of this indeformable part so that relative displacements between these nodes are worthless and one impose possibly displacements on the values resulting from the translation and/or rotation. These nodes are defined by groups of meshes and groups of nodes to which they belong. For the restrictions of use, to see the §3.2.1.
In small disturbances, one imposes in 2D \( (\text{nb}_{\text{ddl}} \times \text{nb}_{\text{nœud}} - 3) \) relations and in 3D \( (\text{nb}_{\text{ddl}} \times \text{nb}_{\text{nœud}} - 6) \) relations, where \( \text{nb}_{\text{ddl}} \) is the number of degrees of freedom per node and \( \text{nb}_{\text{nœud}} \) is the number of nodes of the list given afterwards LIAISON_SOLIDE. One solid is determined by the position of one of its points and a reference mark in this point. Relations are written by taking the vectorial formula translating a rigid movement of body into small rotations:

\[
\vec{u}(M) = \vec{u}(A) + \vec{\Omega}(A) \times \vec{AM}
\]

where \( A \) is an arbitrary node of the solid. For more details, see Doc. [R3.03.02]. It is possible to use LIAISON_SOLIDE in great transformations in STAT_NON_LINE (since the problem becomes non-linear) only while imposing that the loading is following (TYPE_CHARGE='SUIV'), to see [U4.51.03]).

- ♦ Topological assignment: ALL, GROUP_MY, SANS_GROUP_MY, SANS_GROUP_NO
  The conditions kinematics of rigid body are imposed on the nodes belonging to the meshes given by the keywords ALL, GROUP_MA while possibly excluding thanks to the keywords SANS_*.

- ◊ DIST_MIN
  This keyword is used to define a distance (in the units of the grid) below which one considers that the points of the grid are confused. This distance is also used to determine points so are aligned, i.e. if they are in a cylinder of diameter lower than \( d_{\text{min}} \). By default, \( d_{\text{min}} = 0.001 \times \text{armin} \), where \( \text{armin} \) is the smallest edge of the grid.

Note:
If an element has all its nodes in a zone "solidified" by LIAISON_SOLIDE, its deformation is worthless. The state of stress is then also null as well as the generalized efforts if it is about an element of structure.

If the list of the nodes with to bind comprise only aligned nodes port and no degree of freedom of rotation, then, if the system must be subjected to one rotation, it is necessary to add a fictitious node not aligned in the list. The other solution is to place itself on the assumption great rotations and large displacement \( S \) (TYPE_CHARGE='SUIV'), one imposes indeed explicitly with the solid connection to check that the distances are constant.

### 5.17 Keyword LIAISON_ELEM

| LIAISON_ELEM =_F( | ♦ GROUP_MA_1 = lhma1, [l_gr_maille] |
| / GROUP_NO_2 = lhma2, [gr_nœud] |
| / GROUP_MA_2 = lhma2, [l_gr_maille] |
| / OPTION = '/3D_POU', |
| '/3D_TUYAU', |
| '/3D_POU_ARLEQUIN', |
| '/COQ_POU', |
| '/COQ_TUYAU', |
| '/PLAQ_POUT_ORTH', |
| '/2D_POU', |
| ◊ NUME_LAGR = '/NORMAL', [DEFECT] |
| '/AFTER' |
| / ANGL_MAX = /1., [DEFECT] |
| /angl, [R] |

If option == '3D_POU_ARLEQUIN' ♦ CARA_ELEM = will cara, [cara_elem] ♦ CHAM_MATER = to subdue, [cham_mater] If option == 'PLAQ_POUT_ORTH' ◊ EXCENT_POUTRE = '/YES', [DEFECT] '/NOT', [DEFECT]
If option == 'COQ_POU' or option == 'COQ_TUYAU' or option == '3D_TUYAU'
◊ AXE_POUTRE = (X, there, Z),
[l_R] ◊ CARA_ELEM = will cara,
[cara_elem] )

The keyword LIAISON_ELEM allows to connect pieces of structure of different modelings. EN appealing “massive part” a piece of structure modelled with isoparametric elements 3D, this keyword factor makes it possible to model the connection:

- Of a massive part with a part beam [R3.03.03] or a pipe section [R3.08.06];
- Of a hull part with a part beam [R3.03.06] or a pipe section [R3.08.06];

This keyword also makes it possible to connect the edge of a structure 2D with a beam or a discrete element.

The goal of this functionality is not to give an account of the scales length between the parts to be connected but to allow a simplification of modeling by replacing a massive or surface part by a beam part for example. The connection is treated by forcing linear relations between the degrees of freedom of the nodes of the junction of the two parts to be connected, without imposing superfluous relations.

◊ NUME_LAGR
See § 4.4.

5.17.1 Option ‘3D_POU’

This option makes it possible to connect a massive part 3D with a part modelled with beams of Euler or Timoshenko. UN connection between a massive part 3D and a beam part requires six linear relations.

◊ Topological assignment
GROUP_MA_1 :
Itte operand definiteT surface meshes of the part massive modelling the trace of the section of the beam on this massive part. These meshes must be affected by finite elements of faces of elements 3D before. The massive part must be with a grid with quadratic elements because the coefficients of the relations to be imposed are numerically integrated geometrical quantities. So that these integrals are evaluated correctly, it is necessary to have quadratic elements.

GROUP_NO_2, GROUP_MA_2 :
These operands define the node of beam to connect to the massive part. Thus if one uses GROUP_NO_2, one should give one group, this one containing one node.
If one uses GROUP_MA_2, it is necessary that the mesh is single and of type POI1.

◊ ANGL_MAX
Give the angle (in degree) allowing to check if the meshes of list GROUP_MA_1 normals have forming an angle higher than ANGL_MAX between them. If it is the case, there is emission of a message of alarm.

5.17.2 Option ‘3D_POU_ARLEQUIN’

This option makes it possible to connect a massive part 3D with a part modelled with beams of Timoshenko within the framework Harlequin.

◊ Topological assignment
GROUP_MA_1 :
These operands define the voluminal meshes of the part massive included in the zone of covering. These meshes must be affected by voluminal elements 3D before. The massive part can be with a grid with linear or quadratic elements.
GROUP_MA_2 :

These operands define the meshes of the type beam to connect to the massive part. These meshes must be affected by elements 1D of beams of Timoshenko before.

♦ CARA_ELEM

Concept created by the order AFFE_CARA_ELEM, containing the geometrical characteristics of the beam, being used with construction of the matrices as coupling Harlequin.

♦ CHAM_MATER

Concept created by the order AFFE_MATERIAU, containing characteristic materials of the beam, being used with construction of the matrices as coupling Harlequin. These characteristics are supposed being balanced by the user, within the meaning of the partition of the unit necessary to the framework Harlequin.

5.17.3 Option '2D_POU'

This option makes it possible to connect a surface part 2D to a part modelled with a beam of Euler or discrete.

♦ Topological assignment

GROUP_MA_1 :

Ittte operand definiteT meshes of edge of the part 2D to connect to the element 1D. The surface part must be with a grid with quadratic elements because the coefficients of the relations to be imposed are numerically integrated geometrical quantities. So that these integrals are evaluated correctly, it is necessary to have quadratic elements.

GROUP_NO_2, GROUP_MA_2 :

These operands define the node of beam to connect to the massive part. SI one uses GROUP_NO_2, one should give one group, this one containing one node, SI one uses GROUP_MA_2, it is necessary that the mesh is single and of type POI1.

5.17.4 Option 'COQ_POU'

This option makes it possible to connect a part with a grid in hull with a beam part. The trace of the section of the beam on the hull part must correspond exactly to the meshes of edge defined by GROUP_MA_1. This implies the identity of the centres of inertia, of surfaces of the sections hull and beam in opposite.

♦ Topological assignment

GROUP_MA_1 :

Ittte operand definiteT meshes of edge of the part with a grid in hulls (the meshes of edge are thus SEG2 or SEG3 according to selected modeling). These meshes must be affected by finite elements of edge of hulls before.

GROUP_NO_2 :

Ittte operand defines the node of beam to connect to the hull part. ON should give one group, this one containing one node.

♦ AXE_POUTRE

Allows to define the axis of the beam to be connected, whose end is lno2 or lgno2 (1 only node).

♦ CARA_ELEM

Concept created by the order AFFE_CARA_ELEM, containing the geometrical characteristics of the hull.

5.17.5 Option '3D_TUYAU'

This option makes it possible to connect a massive part 3D with a part modelled with elements pipe. A connection between a massive part 3D and a pipe part requires six linear relations for the degrees of
freedom of beam, plus a relation on the mode of swelling, plus twelve relations corresponding to the transmission of the modes of Fourier two and three of ovalization of the pipe.

- **Topological assignment**

  **GROUP_MA_1**:
  
  Ittte operand defines the surface meshes of the part massive modelling the trace of the section of the pipe on this massive part. These meshes must be affected by finite elements of faces of elements 3D before.

  **GROUP_NO_2**:
  
  Ittte operand defines the node of pipe to connect to the massive part. ON should give one group, this one containing one node.

- **AXE_POUTRE**

  Defines the axis of the pipe to connect, whose end is only one node (lgno2).

- **CARA_ELEM**

  Concept created by the order AFFE_CARA_ELEM, containing the geometrical characteristics of the pipe.

- **ANGL_MAX**

  Give the angle (in degree) allowing to check if the meshes of list GROUP_MA_1 normals have forming an angle higher than ANGL_MAX between them. If it is the case, there is emission of a message of alarm.

### 5.17.6 Option ‘COQ_TUYAU’

This option makes it possible to connect a part with a grid in hull to a part with a grid with elements pipe. The trace of the section of the pipe on the hull part must correspond exactly to the meshes of edge defined by GROUP_MA_1. This implies the identity of the centres of inertia, of surfaces of the sections hull and pipe in opposite. **Consequently connections of type “pricking” are impossible.** A connection between a hull part and a pipe part requires the same linear relations as the option COQ_POU on the degrees of freedom of beam of the element pipe besides the relations on the degrees of freedom of ovalization, of warping and swelling.

- **Topological assignment**

  **GROUP_MA_1**:
  
  Ittte operand defines the meshes of edge of the part with a grid in hulls (the meshes of edge are thus SEG2 or SEG3 according to selected modeling). These meshes must be affected by finite elements of edge of hulls before.

  **GROUP_NO_2**:
  
  Ittte operand defines the node of pipe to connect to the massive part. ON should give one group, this one containing one node.

- **AXE_POUTRE**

  Defines the axis of the pipe to connect, whose end is only one node (lgno2).

- **CARA_ELEM**

  Concept created by the order AFFE_CARA_ELEM, containing the geometrical characteristics of the pipe and the hull.
5.17.7 Option ‘PLAQ_POUT_ORTH’

This option makes it possible to connect a part with a grid with elements TRI3 and QUA4 (modelings DKT, DST and DKTG) with a part modelled by an element of beam or discrete. With an aim of simplifying the entry of the data the following checks are not carried out:

- There is no checking which the axis of the beam is perpendicular to the plate;
- There is no checking between the calculation of the mechanical characteristics (S, I,...) realized on the meshes of the trace of the section of beam and the mechanical characteristics assigned to the beam with the assistance CARA_ELEM.

♦ Topological assignment

GROUP_MA_1 :

Ittte operand defines the meshes of plate who model the trace of the section of the beam on this part. These meshes must be affected by finite elements of plate, modelings DKT, DST and DKTG.

GROUP_NO_2 :

Ittte operand defines the node of beam to connect to the plate. ON should give one group, this one containing one node.

◊ ANGL_MAX

Give the angle (in degree) allowing to check if the meshes DE list GROUP_MA_1 normals have forming an angle higher than ANGL_MAX between them. If it is the case, there is emission of a message of alarm.

◊ VERIF_EXCENT = ‘NOT’/‘YES’

The node of the beam must coincide, except for a tolerance, with the centre of gravity of the meshes which model the trace of this beam on the flagstone. In the event of nonrespect from this rule, two behaviors are possible:

- If VERIF_EXCENT= OUI’, behavior by default, an error message is emitted and the code stops in fatal error;
- If VERIF_EXCENT= NON’, a message of information is transmitted.

This operand makes it possible not to be obliged to position exactly the beams in the centre of gravity of the trace of the section, which is not inevitably known at the time of the realization of the grid. In the case, where this rule is not complied with, the user is informed of the distance between node beam and this centre of gravity are by a fatal error (VERIF_EXCENT=’ OUI’ ) maybe by the emission of a message of information (VERIF_EXCENT=’ NON’).

5.18 Keyword LIAISON_RBE3

The keyword LIAISON_RBE3 D allowsE to define linear relations of type RBE3 between the degrees of freedom of a main node and nodes slaves. They are relations making it possible to specify the value of certain degrees of freedom of a main node as being the weighted average of certain displacements and certain rotations of nodes slaves.

The produced linear relations are such as the efforts seen by the main node are distributed to the nodes slaves proportionally at their distance to the centre of gravity of the nodes slaves. The possible

2 To make these checks would be needed that the user gives besides the node of the beam, the name of mesh affected by CARA_ELEM who has as an end the node of connection. In the majority of the cases this mesh is unknown of the user, it is the software of grid which defines its name.
additional weightings provided by the user can be taken into account. For more precise details, one will be able to refer to Doc. of reference [R3.03.08].

This option allows raccommoder a massive part 3D with a part modelled with beams of Euler or Timoshenko. UN connection between a massive part 3D and a beam part requires six linear relations.

♦ **Topological assignment**

  GROUP_NO_MAIT :
  Identification it main node of the linear relation. ON should give one group, this one containing one node.

  GROUP_NO_ESCL :
  Identification of the nodes slaves of the linear relation.

♦ **DDL_MAIT**


♦ **DDL_ESCL**

  Identification of the degrees of freedom of the nodes slaves implied in the linear relation. The list must have a length equal to the number of nodes slaves. Each term of the list must be a combination of the entries ‘DX’, ‘DY’, ‘DZ’, ‘DRX’, ‘DRY MARTINI’, ‘DRZ’, separated by an indent ‘-’.

◊ **COEF_ESCL**

  List of weighting coefficients of the terms of the linear relation for each node slave. The list must:
  •  That is to say to have the same length as the number of nodes slaves;
  •  That is to say to be length 1, in which case this coefficient is used for all the nodes slaves.

◊ **NUME_LAGR**

  See § 4.4.

**Example :**

If one wants to create a relation of the type RBE3 between:

- Degrees of freedom ‘DX’, ‘DY’, ‘DZ’, ‘DRX’ main node ‘NO1’;

And:

- Degrees of freedom ‘DX’, ‘DY’, ‘DZ’ node slave ‘NO2’ with the weighting coefficient 0.1;
- Degrees of freedom ‘DX’, ‘DY’, ‘DZ’, ‘DRX’ node slave ‘NO3’ with the weighting coefficient 0.2;
- Degrees of freedom ‘DX’, ‘DY’, ‘DZ’, ‘DRX’ node slave ‘NO4’ with the weighting coefficient 0.3;

One must write the order:

LIAISON_RBE3=_F ( GROUP_NO_MAIT = 'NO1',
                  DDL_MAIT = ('DX', 'DY', 'DZ', 'DRX'),
                  GROUP_NO_ESCL = ('NO2', 'NO3', 'NO4'),
                  DDL_ESCL = ('DX-DY-DZ',
                              'DX-DY-DZ-DRX',
                              'DX-DY-DZ-DRX'),
                  COEF_ESCL = (0.1, 0.2, 0.3),)
6 Loadings of the Dirichlet type for the elements of structure

6.1 Keyword DDL_POUTRE

```
DDL_POUTRE = _F ( ♦ /TOUT = 'YES',
                 /GROUP_NO = lgno, [l_gr_noeud]
                 /GROUP_MA = lgma, [l_gr_maille]
                ♦ SANS_GROUP_MA = lgma1, [l_gr_maille]
                ♦ SANS_GROUP_NO = lgno1, [l_gr_noeud]
                ♦ |DX = ux, [R]
                    |DY = uy, [R]
                    |DZ = uz, [R]
                    |DRX = drx, [R]
                    |DRY MARTINI = dry Martini, [R]
                    |DRZ = drz, [R]
                ♦ GROUP_MA_REPE = lgma, [l_gr_maille]
                ♦ /ANGL_VRIL = G, [R]
                ♦ /VECT_Y = (V1, V2, V3) [ l_R]
             )
```

The keyword **DDL_POUTRE** allows blocking of degrees of freedom in a local reference mark of a beam. The local reference mark of a beam is defined:

- By the axis \( X \) determined by the mesh to which the node belongs. The mesh is directed towards the specified node. To avoid the indetermination, it is necessary that the node to which the condition relates belongs to only one \( SEG \). In the case or it belongs to several meshes, the user defines the mesh giving the local orientation.
- By **VECT_Y**, a vector of which projection on the orthogonal level with the axis \( X \) the axis defines \( Y \). The axis \( Z \) is given using \( X \) and \( Y \).
- By **ANGL_VRIL**, an angle of gimlet, given in degrees, makes it possible to direct a local reference mark around the axis \( X \).

- **Topological assignment:** ALL, GROUP_MY, GROUP_NO, SANS_GROUP_MY, SANS_GROUP_NO
  The conditions kinematics are imposed on the nodes given by the keywords ALL, GROUP_MY, GROUP_NO while possibly excluding thanks to the keywords SANS_.*

- **Components:** \( ux, uy, uz, drx, dry Martini, drz \)
  See their significance § 5.1.

- **GROUP_MA_REPE**
  Definition of the reference mark of the beam on the last mesh.

- **ANGL_VRIL**
  Angle of gimlet, given in degrees, makes it possible to direct a local reference mark around the axis \( X \).

- **VECT_Y**
  Vector of which projection on the orthogonal level with the axis \( X \) the axis defines \( Y \). The axis \( Z \) is given using \( X \) and \( Y \).

6.2 Keyword LIAISON_COQUE

```
LIAISON_COQUE = _F ( ♦ | GROUP_MA_1 = l_gma1, [l_gr_maille]
                     | GROUP_NO_1 = l_gno1, [l_gr_noeud]
                    ♦ SANS_GROUP_MA_1 = lgma1, [l_gr_maille]
                    ♦ SANS_GROUP_NO_1 = lgno1, [l_gr_noeud]
                    ♦ | GROUP_MA_2 = l_gma2, [l_gr_maille]
                        | GROUP_NO_2 = l_gno2, [l_gr_noeud]
                    ♦ SANS_GROUP_MA_2 = lgma2, [l_gr_maille]
                    ♦ SANS_GROUP_NO_2 = lgno2, [l_gr_noeud]
                 )
```

**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 2020 EDF R&D - Licensed under the terms of the GNU FDL (http://www.gnu.org/copyleft/fdl.html)
LE keyword LIAISON_COQUE allows to represent the connection enters of the hulls by means of relations flaxéaires. The classical approach admits that two plans with a grid in hulls are cut according to a line which belongs to the grid of the structure. Couples of nodes being obtained while putting in opposite two lists of nodes (see § 5.3.1). That has the disadvantage of twice counting the volume which is the intersection of the two hulls.

The idea is thus to stop the grid of a hull perpendicular to a hull given to the level of the higher or lower skin of the latter.

One represented in features full volume with the hulls and in dotted lines the average plans of these hulls (which result from the grid). The horizontal hull stops in $A_1A_2$ and the projection of $A_1A_2$ on the average level of the vertical hull is $B_1B_2$ (that one represented in full features). The link between the two hulls is made by connections of solid body (see § 5.16) between the nodes in with respect to the segments $A_1A_2$ and $B_1B_2$.

For example for the nodes $A_1$ and $B_1$, one will write the formula (valid in small rotations):

$$U(B_1)=U(A_1)+\Omega(A_1)\wedge A_1B_1$$  \hspace{1cm} (7)

And equality of rotations:

$$\Omega(B_1)=\Omega(A_1)$$  \hspace{1cm} (8)

**Topological assignment**

GROUP_MA_1, GROUP_NO_1 :

First list of nodes to be connected. nodes of which given by the keywords GROUP_MA_1 , GROUP_NO_1 while possibly excluding thanks to the keywords SANS_* . These nodes represent the trace of the hull perpendicular to the current hull. On our example, they would be the nodes of the segment $B_1B_2$ or of the segment $A_1A_2$.

GROUP_MA_2, GROUP_NO_2 :

Second list of nodes to be connected. nodes of which given by L be keywords GROUP_MA_2 , GROUP_NO_2 while possibly excluding thanks to the keywords SANS_* . These nodes belong with the perpendicular hull and in the nodes of the first list. Opposite is adjusted by the program according to the criterion of smaller distance. On our example if the first list is drawn up by the nodes of $A_1A_2$, the second list is drawn up by the nodes of $B_1B_2$.

◊ NUME_LAGR

See § 4.4.
7 Loadings of the Neumann type

7.1 Keyword FORCE_NODEALE

```
FORCE_NODEALE=_F ( ♦ GROUP_NO = lgno, [l_gr_noeud]
   ♦ | FX  = fx, [R] or [function]
   ♦ | FY  = fy, [R] or [function]
   ♦ | FZ  = fz, [R] or [function]
   ♦ | MX  = MX, [R] or [function]
   ♦ | MY  = my, [R] or [function]
   ♦ | MZ  = mz, [R] or [function]
   ◊ ANGL_NAUT = (has, B, G) [l_R] or [l_fonction]
)
```

The keyword factor FORCE_NODEALE is usable to apply, with nodes or groups of nodes, nodal forces, definite component by component in the reference mark total or in a reference mark oblique defined by three nautical angles. In any rigour, the application of a nodal loading is physically incorrect and can cause stress concentrations. It is preferable to use loadings distributed.

♦ Topological assignment: GROUP_NO
The loading is affected on the nodes.

♦ Components: fx, fy, fz, MX, my, mz
Values of the components of the nodal forces of the nodal moments applied to the specified nodes. These nodal forces will come to be superimposed on the nodal forces resulting, possibly, other loadings. Into axisymmetric, the values correspond to a sector of a radian (to divide the real loading by \( 2\pi \)).

◊ ANGL_NAUT
List of the three angles, in degrees, which define the oblique reference mark of application of the nodal forces (the last angles of the list can be omitted if they are worthless). The nautical angles make it possible to pass from the total reference mark of definition of the coordinates of the grid to an unspecified oblique reference mark (see §3.9.3). By defaults the angles are identically worthless and thus the components of forces are defined in the reference mark total.

7.2 Keyword FORCE_ARETE

```
FORCE_ARETE=_F ( ♦ GROUP_MA = lgma, [l_gr_maille]
   ♦ | FX  = fx, [R] or [function]
   ♦ | FY  = fy, [R] or [function]
   ♦ | FZ  = fz, [R] or [function]
   ♦ | MX  = MX, [R] or [function]
   ♦ | MY  = my, [R] or [function]
   ♦ | MZ  = mz, [R] or [function]
)
```

The keyword factor FORCE_ARETE is usable to apply forces linear, with one edge of element voluminal or of hull, definite component by component in the reference mark total. This edge is defined by one or more meshes or of the groups of meshes of the type segment.

♦ Topological assignment: GROUP_MY
The loading is affected on the meshes which are necessarily segments.

♦ Components: fx, fy, fz, MX, my, mz
Values of the components of the forces and the linear moments applied to the meshes specified. This loading applies to following modelings: DKT, DST, Q4G, 3D and COQUE_3D.
7.3 **Keyword FORCE_CONTOUR**

\[
\text{FORCE\_CONTOUR}=\_F \quad (\quad \star \quad \text{GROUP\_MA} = \text{lgma}, \quad [l\_gr\_maille] \\
\quad | \quad FX = \text{fx}, \quad [\text{R}] \text{ or } [\text{function}] \\
\quad | \quad FY = \text{fy}, \quad [\text{R}] \text{ or } [\text{function}] \\
\quad | \quad FZ = \text{fz}, \quad [\text{R}] \text{ or } [\text{function}] 
}\]

The keyword factor **FORCE_CONTOUR** is usable to apply forces **linear** with **edge of a 2D field**, definite component by component in the reference mark **total**. This contour is defined by one or more meshes of the type **segment**.

Note: basically it is about a linear force but the unit is a surface force because one reasons for a thickness unit (plane constraints, plane deformations) or divided by \(2\pi\) for axisymmetric modelings (see §3.5).

- **Topological assignment**: **GROUP** _MY_  
  The loading is affected on the meshes which are necessarily segments .

- **Components**: \(fx, fy, fz\)  
  Values of the components of the linear forces applied to the specified meshes. This loading applies to following modelings: _D_ _PLAN, AXIS_ and _AXIS_FOURIER_, including XFEM.

7.4 **Keyword FORCE_FACE**

\[
\text{FORCE\_FACE}=\_F \quad (\quad \star \quad \text{GROUP\_MA} = \text{lgma}, \quad [l\_gr\_maille] \\
\quad | \quad FX = \text{fx}, \quad [\text{R}] \text{ or } [\text{function}] \\
\quad | \quad FY = \text{fy}, \quad [\text{R}] \text{ or } [\text{function}] \\
\quad | \quad FZ = \text{fz}, \quad [\text{R}] \text{ or } [\text{function}] 
}\]

The keyword factor **FORCE_FACE** is usable for to apply forces **surface** (thus of the dimension of a pressure) on one **face of element 3D**, definite component by component in the reference mark **total**. This face is defined by one or more meshes of the type **triangle** or **quadrangle**.

- **Topological assignment**: **GROUP** _MY_  
  The loading is affected on the meshes which are necessarily triangles or of the quadrangles .

- **Components**: \(fx, fy, fz\)  
  Values of the components of the surface forces applied to the meshes specified. This loading applies to following modelings: _3D, 3D_HHM, 3D_HM, 3D_THHM, 3D_THM, 3D_HH2, 3D_THH2M_ and XFEM .

7.5 **Keyword FORCE_INTERNE**

\[
\text{FORCE\_INTERNE}=\_F \quad (\quad \star \quad \text{GROUP\_MA} = \text{lgma}, \quad [l\_gr\_maille] \\
\quad | \quad FX = \text{fx}, \quad [\text{R}] \text{ or } [\text{function}] \\
\quad | \quad FY = \text{fy}, \quad [\text{R}] \text{ or } [\text{function}] \\
\quad | \quad FZ = \text{fz}, \quad [\text{R}] \text{ or } [\text{function}] 
\}
\]

The keyword factor **FORCE_INTERNE** is usable in two cases:

- To apply forces **voluminal** on one **field 3D**, definite component by component in the reference mark **total**. This field is defined by one or more meshes of the type **hexahedron**, **tetrahedron**, **pyramid** or **pentahedron** .

- To apply forces **voluminal** on one **2D field**, definite component by component in the reference mark **total**. This field is defined by one or more meshes of the type **triangle** or **quadrangle** .

- **Topological assignment**: **ALL, GROUP** _MY_
The loading is affected on the meshes which are necessarily triangles or of the quadrangles in 2D and the hexahedrons, tetrahedrons, pyramids or pentahedrons in 3D.

- Components: fx, fy, fz
  Values of the components of the voluminal forces applied to the specified meshes.

For the case 3D, this loading applies to following modelings: 3D, 3D_SI, 3D_INCO, 3D_HHMD, 3D_HMD, 3D_THHD, 3D_THHMD, 3D_THMD, 3D_THM, 3D_HM, 3D_THH, 3D_HHMD.

For the case 2D, this loading applies to following modelings: C_PLAN, D_PLAN, AXIS, AXIS_FOURIER, AXIS_SI, AXIS_INCO, AXIS_THHM, AXIS_HM, AXIS_TH, AXIS_HH, AXIS_THM, D_PLAN_THHM, D_PLAN_HM, D_PLAN_THH, D_PLAN_HHMD, D_PLAN_THM.

### 7.6 Keyword PRES_REP

```
PRES_REP  =_F ( ♦ / ALL  = 'YES',
             / GROUP_MA  = lgma,  [l_gr_maille]
             | CRACK  = fiss,  [fiss_xfem]
             ♦ | NEAR  = 'P',  [R] or [function]
             | CISA_2D  = 'T',  [R] or [function]
        ),
```

The keyword factor PRES_REP is usable to apply one pressure with a field of continuous medium 2D or 3D, one pressure on a hull of the type COQUE_3D or one shearing with a field of continuous medium 2D.

- Topological assignment: ALL, GROUP_MY
  The loading is affected on the meshes which are necessarily segments in 2D and triangles or quadrangles in 3D.

- CRACK
  The imposition of a pressure on the lips of a crack X-FEM is done by the specific keyword CRACK, since no group of mesh corresponds to the lips. One informs the names of the cracks then (coming from the order DEFI_FISS_XFEM [U4.82.08]) to which one wishes to apply the pressure. Attention, it is not possible to apply a loading of pressure to the lips of a model XFEM of the cohesive type.

- NEAR
  Value of the imposed pressure. The pressure is Positive according to the contrary direction of the normal to the element. That is to say the tensor of the constraints, the imposed loading is:
  \[ \sigma_{ij} n_i n_j = -p n_i n_j \]. This loading applies to following modelings: AXIS, D_PLAN, C_PLAN, AXIS_FOURIER, D_PLAN_HHM, D_PLAN_HM, D_PLAN_THHM, D_PLAN_THM, AXIS_HH, AXIS_THHM, AXIS_TH, TUYAU_3M, TUYAU_6M, 3D_HHM, 3D_HM, 3D_THH, 3D_THHMD, 3D_THM, 3D_HM, 3D_THH, 3D_HHMD, 3D_THM, and COQUE_3D.

In the case of a pressure "function", the dependence with the geometry can be made compared to the initial geometry with the parameters X, Y, and Z or compared to the reactualized geometry (only if TYPE_CHAR = ‘SUIV’) with the parameters XF, YF, and ZF.

Note:
The pressures of type “function”, are not for the moment not compatible with modeling COQUE_3D.

- CISA_2D
  Value of imposed shearing. Shearing Epositive St following the tangent to the element. This loading applies to following modelings: AXIS, D_PLAN, C_PLAN and AXIS_FOURIER.
7.7 **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 pressure (correspondent to a loading of the type PRES_REP), densities of voluminal force in 2D or 3D (correspondent with a loading of the type FORCE_INTERNE) and of the densities of surface force in 2D or 3D (correspondent with a loading of the type FORCE_FACE and FORCE_CONTOUR).

These loadings are always of “following” type (see §3.6.2).

7.8 **Keyword EFFE_FOND**

```
EFFE_FOND = _F(
    GROUP_MA = lgma,
    GROUP_MA_INT = gtrou,
    NEAR = P,
),
```

The keyword factor EFFE_FOND is usable for to calculate the basic effect on a branch of piping (modeling 3D exclusively) subjected to an internal pressure $P$.

- **Withtopological ffection:** GROUP_MY
  Together surface meshes (triangles or quadrangles) modelling the material section of piping (gmat on the figure) where the pressure will be applied. This surface must be directed according to same convention as for the loading PRES_REP (cf §7.6).

- **GROUP_MA_INT**
  Inseeem linear meshes (segments) modelling the contour of the hole (gtrou on the figure). The knowledge of these meshes is necessary because one needs to calculate the surface of the hole. Indeed, the effort resulting (or basic effect) due to stopping from the hole at the end is worth:

  $$ F_n = \pi R_i^2 P $$

  (9)

  This basic effort or effect applies to the wall of the tube (gmat). The effort divided correspondent is worth then:
Pressure interns with piping. One applies ruffle in practice $F_p$ with $g_{mat}$ (a value positive correspondent for an effort directed in the contrary direction with the outgoing normal and a basic purpose inducing a traction on piping).

### 7.9 Keyword GRAVITY

**Keywords:**
- $\text{GRAVITY} = _F(\diamond) \text{ GROUP\_MA} = \text{lgma, } [l\_gr\_maille]$
- $\diamond / \text{REVOLVE} = G, [R]$
- $\diamond / \text{DIRECTION} = (h_{sp}, B_p, C_p), [l\_R]$

The keyword factor **GRAVITY** is usable for to apply a field of gravity to the model. $g$ represent the intensity of the field of gravity and $(a_p, b_p, c_p)$ specify the direction and the direction of application of the field. The loading which results from it is form:

$$F_p = \frac{\pi R_i^3}{\pi [R^2 - R_i^2]} P = \rho g S_{mat} S_{mm}$$

(10)

- NEAR

7.10 Keyword ROTATION

**Keywords:**
- $\text{ROTATION} = _F(\diamond) / \text{ALL} = \text{'YES'},$
- $/ \text{GROUP\_MA} = \text{lgma, } [l\_gr\_maille]$
- $\diamond \text{SPEED} = \text{Omega}, [R]$
- $\diamond \text{AXIS} = (h_{sr}, B_r, C_r), [l\_R]$
- $\diamond \text{CENTER} = (X, there, Z), [l\_R]$

The keyword factor **ROTATION** is usable for to apply a field of force are equivalent to the centrifugal force applying to a structure in rotation. That is to say $\omega$ number of revolutions and $(a_r, b_r, c_r) \ L$ * axis of rotation.

$$\omega = \frac{\omega [a_r i + b_r j + c_r k]}{\sqrt{a_r^2 + b_r^2 + c_r^2}}$$

(12)

The loading which results from it is form:
\[ f = \rho |\omega \wedge OM| \wedge \omega \]  

(13)

Where \( O \) is the origin of the coordinates and \( M \) a point running of the structure with \( \rho \) definite density like characteristic of material (see operators DEFI_MATERIAU [U4.43.01] and AFFE_MATERIAU [U4.43.03]).

There can be only one occurrence of this keyword in AFFE_CHAR_MECA.

- Topological assignment: ALL, GROUP_My
- The loading is affected on these meshes.
- SPEED
- Number of revolutions.
- AXIS
  - Axis of rotation. For plane modelings, the axis of rotation must be in the direction \( Oz \) and for axisymmetric modelings and Fourier, it must be in the direction \( Oy \).
- CENTER
  - If the centre of rotation is not the origin (defect), one can specify his coordinates \((X, \text{there}, Z)\). For axisymmetric modelings and Fourier, the center must be the origin.

One can vary in time the number of revolutions by breaking up rotation in a multiplicative way between space loading and evolution into time \( \omega(t) = \omega_0 f(t) \), then while multiplying load by a multiplying function (keyword FONC_MULT) in transitory calculation. However, it is advisable to pay attention: the loading \( \rho |\omega \wedge OM| \wedge \omega \) being proportional to the square number of revolutions, \( \omega(t)^2 \), it is necessary to affect the square of the evolution in time, \( f(t)^2 \), behind FONC_MULT.

### 7.11 Keyword PRE_SIGM

PRE_SIGM

The keyword factor PRE_SIGM is usable to apply a prestressing \( \sigma_{\text{pre}} \). This loading makes it possible to apply average voluminal constraints, overall uniform (2D or 3D) with a voluminal field. The second calculated elementary member will be:

\[ \int_{V_e} \sigma_{\text{pre}} : \varepsilon (v^*) dV_e \]  

(14)

This loading is known under the option FORC_NODA that one finds in the order CALC_CHAMP or during the phase of prediction of Newton of the operator STAT_NON_LINE.

The stress field \( \sigma_{\text{sigm}} \) is of type map or cham_elga. It can come from CREA_CHAMP or to be calculated in addition.

One should not confuse this prestressing with the initial constraint \( \sigma_{\text{ini}} \) used into nonlinear, because this prestressing does not intervene directly in the expression of the law of behavior. This field of prestressing is used like second member in the resolutions of MECA_STATIQUE and STAT_NON_LINE.

### 7.12 Keyword PRE_EPSI

PRE_EPSI = \_F( ♦ / ALL = 'YES', ♦ / GROUP_MA = lgma, [l_gr_maille]
♦ | EFXX = epsxx, [R] or [function]
♦ | EFYY = epsyy, [R] or [function]
♦ | EFZZ = epszz, [R] or [function]
♦ | EFXY = epsxy, [R] or [function]
♦ | EFXZ = epsxz, [R] or [function]
♦ | EFYX = epsyz, [R] or [function]
♦ | EFZX = epsxz, [R]
♦ | KY = ky, [R]
♦ | KZ = kz, [R]

---

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 2020 EDF R&D - Licensed under the terms of the GNU FDL (http://www.gnu.org/copyleft/fdl.html)
The keyword factor `PRE_EPSI` is usable to apply one predeformation \( \varepsilon_{\text{pre}} \). It is a loading of deformation average, overall uniform applied to an element 2D, 3D or of structure. The second calculated elementary member will be:

\[
\int_{V_e} A \varepsilon_{\text{pre}} : \varepsilon(v^*) dV_e
\]  

(15)

where \( A \) indicate the tensor of elasticity (recovered in the field material for all the laws for which are defined the elastic characteristics).

One should not confuse this predeformation with the initial deformation \( \varepsilon_{\text{ini}} \) used EN nonlinear, because this predeformation does not intervene directly in the expression of the law of behavior.

This predeformation is usable for example to solve the elementary problems determining the elastic correctors in the basic cell (2D, 3D), in periodic homogenisation. The moduli of homogenized elasticity are obtained while calculating by the operator `POST_ELEM` [U4.81.22] keyword `ENER_POT` potential energy of elastic strain to balance starting from the correctors. But that can be useful for other applications.

- **Topological assignment:** ALL, GROUP_MY
  The loading is affected on these meshes.

- **Components:** `epsxx`, `epsyy`, `epszz`, `epsxy`, `epsxz`, `epsyz`
  Values of the components of the tensor of the initial deformations in the total reference mark for the isoparametric elements 2D or 3D ( `C_PLAN`, `AXIS`, `D_PLAN`, `3D`, `3D_SI`, `AXIS_SI`, `D_PLAN_SI` )

- **Component:** `epsx`
  Constant value by element of the elongation according to the local axis of the beam ( `POU_D_E`, `POU_D_T`, `POU_D_TG` ).

- **Component:** `ky`
  Constant value by element of the variation of curve according to the axis \( y \) room \(- \frac{d \theta_y}{dx}\) beam ( `POU_D_E`, `POU_D_TG` ).

- **Component:** `kz`
  Value constant by element variation of curve according to the axis \( z \) room \( \frac{d \theta_z}{dx}\) beam ( `POU_D_E`, `POU_D_TG` ).

- **Components:** `exx`, `eyy`, `exy`
  Constant values by element of the deformations of membrane in the local reference mark hull ( `DKT`, `DST`, `Q4G` ).

- **Components:** `kxx`, `kyy`, `kxy`
  Constant values by element of the variations of curve in the local reference mark of the hull ( `DKT`, `DST`, `Q4G` ).
7.13 Keyword FORCE_ELEC

The keyword factor FORCE_ELEC is usable to apply the force of Laplace acting on a principal driver, due to the presence of one secondary driver right (not being based on part of grid) compared to this principal driver. When the secondary driver is not right, the keyword will be used INTE_ELEC (see §7.14).

In fact, the loading defined by FORCE_ELEC a module has which must be multiplied by the temporal function of intensity specified by the operator DEFI_FONC_ELEC [U4.MK.10] to really represent the force of Laplace.

The function of space composing the linear density of force of Laplace exerted in a point $M$ driver 1 (principal driver) by the elements of the driver 2 (secondary driver) is:

$$f(M) = \frac{e_1 \wedge e_2 \wedge r}{2} \int_2 \frac{e_2 \wedge r}{\|r\|^3} ds_2$$

Of which here the chart:

In the case of a secondary right and finished driver, this expression becomes:

$$f(M) = \frac{e_1 \wedge n}{2} \left| \sin \alpha_1 - \sin \alpha_2 \right|$$

Of which here the chart:

In the typical case of the secondary driver infinite right, $\alpha_1$ and $\alpha_2$ tend towards $\frac{\pi}{2}$, one has then:

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 2020 EDF R&D - Licensed under the terms of the GNU FDL (http://www.gnu.org/copyleft/fdl.html)
\[ f(M) = e_1 \wedge \frac{n}{d} \] (18)

- **Topological assignment:** ALL, GROUP_MY
  The principal driver is based on whole or part of the grid made up of linear elements in space.

- **Components:** \( fx, F \) there, \( fz \)
  In this case where there are several secondary drivers infinite and parallel with the principal driver (keywords \texttt{COUR_PRIN} and \texttt{COUR_SECO} in the order \texttt{DEFI_FONC_ELEC}) the components directly are specified \( (fx, fy, fz) \) direction of the force of Laplace who must be normalized with \( 1 \) (either \( fx^2 + fy^2 + fz^2 = 1 \))

- **POSITION = 'PARA'/'FINISHED'/'INFI'**
  Specify the way in which one will define the position of the conducting second.

  - **POSITION = 'PARA'**
    For one secondary driver infinite and parallel with the principal driver, there is at the time two manners of defining the secondary driver.

    - **TRANS** = \( (U_x, U_{there}, U_z) \)
      \( (U_x, U_{there}, U_z) \) will define translation bringing of the principal driver \( 1 \) with the secondary driver \( 2 \).

    - **DIST** = \( D \)
      \( POINT2 = (X_2, there_2, Z_2) \)
      The secondary driver \( 2 \) is defined by its distance \( d \) with principal driver \( 1 \) and a second point \( (X_2, there_2, Z_2) \).

  - **POSITION = 'FINISHED'**
    For a finished secondary driver and not-parallel with the principal driver.

    - **NOT 1 = (X_1, there_1, Z_1)**
      \( POINT2 = (X_2, there_2, Z_2) \)
      The secondary driver \( 2 \) is defined by two points \( (X_1, there_1, Z_1) \) and \( (X_2, there_2, Z_2) \) correspondent at his ends.
It is preferable to choose POINT1 and NOT 2 such as the current circulates of POINT1 with NOT 2.

♦ POSITION = 'INFI'
For an infinite secondary driver and not-parallel with the principal driver.

♦ NOT 1 = (X₁, there₁, Z₁)
  POINT2 = (X₂, there₂, Z₂)
The secondary driver 2 is defined by two points (X₁, there₁, Z₁) and (X₂, there₂, Z₂).

It is preferable to choose POINT1 and NOT 2 such as the current circulates of POINT1 with NOT 2.

7.14 Keyword INTE_ELEC

```
INTE_ELEC = _F ( ♦ / ALL = 'YES',
  ♦ / GROUP_MA = lgma, [l_gr_maille]
  ♦ / GROUP_MA_2= lgma2, [l_gr_maille]
  ♦ / TRANS = (Ux, Uthere, Uz), [l_R]
  ♦ / SYME = (Xo, thereo, Zo, Ux, Uthere, Uz), [l_R] )
```

The keyword factor INTE_ELEC is usable to apply the force of Laplace acting on a principal driver, due to the presence of one secondary driver not-right being based on part of grid, by symmetry or translation compared to the principal driver.

In fact, the loading defined by INTE_ELEC a module has which must be multiplied by the temporal function of intensity specified by the operator DEFI_FONC_ELEC [U4.MK.10] to really represent the force of Laplace.

It is pointed out that the function of space composing the linear density of force of Laplace exerted in a point M driver 1 (principal driver) by the elements of the driver 2 (secondary driver) is:

$$ f(M) = \frac{e_i}{2} \wedge \int_2 e_s \wedge r \, ds_2 $$  \hspace{1cm} (19)

Of which here the chart:
It is the same function as in the case of the keyword `FORCE_ELEC`. On the other hand, for each element \( i \) secondary driver, one calculates his contribution starting from the preceding expression and one summons:

\[
f(M) = \sum_i \frac{e_1 \wedge n}{d} \sin \alpha_1 - \sin \alpha_2
\]

Of which here the chart:

\[
\begin{align*}
\text{with } n &= \frac{e_2 \wedge d}{d}, \quad d = \|d\|, \quad \|d\| = 1
\end{align*}
\]

- **Topological assignment**: `ALL, GROUP_MY`
  - Definition of geometry of the principal driver where the loading is affected.

- **Topological assignment**: `GROUP_MA_2`
  - Definition of the geometry of the secondary driver.

- **TRANS** = \((U_x, U_{there}, U_z)\)
  - \((U_x, U_{there}, U_z)\) will define translation bringing of the principal driver 1 with the secondary driver 2.

- **SYME** = \((X_0, there_0, Z_0, U_x, U_{there}, U_z)\)
  - A symmetry compared to the plan given by the point defines \((X_0, there_0, Z_0)\) and the normal \((U_x, U_{there}, U_z)\), commune with the principal driver and the secondary driver.

### 7.15 Keyword `VECT_ASSE`

`VECT_ASSE = chamno [cham_no_DEPL_R]`

The keyword factor `VECT_ASSE` is usable to apply a second member in the form of one `CHAM_NO`. It `CHAM_NO` is transmitted to these orders via the name of the loading. The field of displacements `chamno` is of type `cham_no`. It can come from `CREA_CHAMP` or to be calculated in addition.
8 Loadings of the Neumann type for the elements of structure

8.1 Keyword FORCE_POUTRE

```
FORCE_POUTRE= _F ( ♦ / ALL = 'YES',
              / GROUP_MA = lgma, [l_gr_maille]
♦ TYPE_CHARGE = /'FORCE', [DEFECT]
              /'WIND',
# if TYPE_CHARGE = 'FORCE'
♦ / | FX = fx, [R] or [function]
      | FY = fy, [R] or [function]
      | FZ = fz, [R] or [function]
      | MX = MX, [R] or [function]
      | MY = my, [R] or [function]
      | MZ = mz, [R] or [function]
          / | NR = N, [R] or [function]
              | VY = vy, [R] or [function]
              | VZ = vz, [R] or [function]
              | MT = MT, [R] or [function]
              | MFY = mfy, [R] or [function]
              | MFZ = mfz, [R] or [function]
              / # if TYPE_CHARGE = 'WIND'
♦ / | FX = fx, [R] or [function]
      | FY = fy, [R] or [function]
      | FZ = fz, [R] or [function]
      / | NR = N, [R] or [function]
          | VY = vy, [R] or [function]
          | VZ = vz, [R] or [function]
      )
```

The keyword factor FORCE_POUTRE is usable to apply forces linear, on elements of type beam defined on all the grid or one or more meshes. The forces are definite component by component, that is to say in the reference mark total, that is to say in the reference mark room element defined by the operator AFFE_CARA_ELEM [U4.42.01].

The force of the type thus is defined WIND. If \( p \) is the pressure exerted by the wind on a normal plane surface with its direction \( \mathbf{v} = [v_x, v_y, v_z] \) the unit vector having the direction and the direction the speed of the wind, and \( d \) the diameter of the cable on which the wind is exerted, then:

\[
F_x = p d v_x \\
F_y = p d v_y \\
F_z = p d v_z
\]  \hspace{1cm} (21)

Let us note that one must remain homogeneous in each occurrence of the keyword factor FORCE_POUTRE: that is to say all the components are defined in the reference mark total that is to say all the components are defined in the reference mark room of definition of the beam.

♦ Topological assignment: ALL, GROUP_MY

The loading is affected on these meshes which are necessarily segments.

♦ TYPE_CHARGE = 'FORCE'/' WIND '

Specify the type of load.

♦ TYPE_CHARGE = 'FORCE'

♦ Components: \( fx, fy, fz, MX, MY, MZ \)

Values of the components of the forces or the moments linear (in the reference mark total) applied to the meshes specified with:

\[ fx, fy, fz \] : forces according to \( x, y \) and \( z \)
• \( m_x, m_y, m_z \) : moments according to \( x, y \) and \( z \)

This loading applies to following modelings: POU_D_T, POU_D_E, POU_D_TGM and POU_D_TG. The moments distributed can be applied only to the right beams to constant section.

♦ Components: \( N, v_y, v_z, M_T, m_{fy}, m_{fz} \)

Values of the components of the efforts generalized (in the reference mark room beam) linear applied to the meshes specified with:
• \( n \) : compression/tractive effort
• \( v_y \) : shearing action according to \( y \)
• \( v_z \) : shearing action according to \( y \)
• \( M_T \) : torque
• \( m_{fy} \) : bending moment according to \( y \)
• \( m_{fz} \) : bending moment according to \( z \)

This loading applies to following modelings: POU_D_T, POU_D_E, POU_D_TGM and POU_D_TG. The moments distributed can be applied only to the right beams to constant section.

♦ TYPE_CHARGE = ‘WIND’

♦ Components: \( f_x, f_y, f_z \)

Values of the components of the forces or the moments linear (in the reference mark total) applied to the meshes specified with:
• \( f_x, f_y, f_z \) : forces according to \( x, y \) and \( z \)

This loading applies to following modelings: POU_D_T, POU_D_E, POU_D_TGM and POU_D_TG.

♦ Components: \( N, v_y, v_z \)

Values of the components of the efforts generalized (in the reference mark room beam) linear applied to the meshes specified with:
• \( n \) : compression/tractive effort
• \( v_y \) : shearing action according to \( y \)
• \( v_z \) : shearing action according to \( y \)

This loading applies to following modelings: POU_D_T, POU_D_E, P

### 8.2 Keyword FORCE_TUYAU

```
FORCE_TUYAU= _F ( ♦ / ALL = ‘YES’,
 / GROUP_MA = lgma, [l_gr_maille]
 ♦ NEAR = P, [R] or [function]
)
```

The keyword factor FORCE_TUYAU is usable to apply a pressure to elements pipe, defined by one or more meshes or of the groups of meshes.

♦ Topological assignment: ALL, GROUP_MY

The loading is affected on these meshes which are necessarily segments.

♦ NEAR

Value of the imposed pressure (real or function). The pressure is positive when the pressure is internal with piping. This loading applies to modelings TUYAU_3M and TUYAU_6M.

### 8.3 Keyword FORCE_COQUE

```
FORCE_COQUE = _F ( ♦ / ALL = ‘YES’,
 / GROUP_MA = lgma, [l_gr_maille]
 ♦ PLAN = '/E-MAIL', [DEFECT]
 / 'MOY',
 / 'INF',
 / 'SUP',
 ♦ / | FX = fx , [R] or [function]
)
```

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.
The keyword factor `FORCE_COQUE` is usable to apply efforts surface on elements of type hull (DKT, DST, Q4G, COQUE_3D...). Operands of `FORCE_COQUE` can be defined:

- In the reference mark total with the keywords FX, FY, FZ, MX, MY and MZ if one wants to impose a torque of the surface efforts by component. The component NEAR impose a pressure normal on surface;
- In a reference mark of reference defined on each mesh with the keywords F1, F2, F3, MF1 and MF2 if one wants to impose a torque of the efforts surface by component; LE reference mark is built around the normal with the element of hull \((z_{ref})\) and of a fixed direction \((x_{ref})\) (for the group of mesh) defined by the keyword ANGL_REP at the same time as the thickness of the hull (see keyword factor HULL of the operator AFFE_CARA_ELEM [U4.42.01]).

It is also possible to define a torque of efforts surface on the average level, lower, higher or grid. If one notes \(d\) offsetting, \(h\) the thickness of the hull, \((F2X, F2Y, F2Z, M2X, M2Y, M2Z)\) the torque of the efforts on the level defined by the user (that is to say excentré) and \((F1X, F1Y, F1Z, M1X, M1Y, M1Z)\) the torque of the efforts in the plan of the grid.

The formulas of passage are the following ones:

- If the plan of calculation is the plan of the grid:
  \[
  F2 = F1 \\
  M2 = M1
  \]

- If the plan of calculation is the excentré average layer:
  \[
  F2 = F1 \\
  M2X = M1X - dx F1Y \\
  M2Y = M1Y + dx F1X
  \]

- If the plan of calculation is the excentré higher layer:
  \[
  F2 = F1 \\
  M2X = M1X - d + \frac{h}{2} F1Y \\
  M2Y = M1Y + d + \frac{h}{2} F1X
  \]

- If the plan of calculation is the excentré lower layer:
  \[
  F2 = F1
  \]
Let us note that one must remain homogeneous in each occurrence of the keyword factor FORCE_COQUE: that is to say all in component of effort in the reference mark total that is to say all in component of effort in the reference mark of definition of the hull.

The pressure applied is positive according to the contrary direction of the normal to the element (defined by the orientation of each mesh (cf [§3.9.1]).

- **Topological assignment:** `ALL , GROUP_ MY`
  - The loading is affected on these meshes which are necessarily triangles or of the quadrangles.
- **NEAR**
  - Value of the imposed pressure (real or function), normal with the hull.
- **Components:** `fx, fy, fz, MX, my, mz`
  - Value of the efforts and the moments in the total reference mark.
- **Components:** `f1, f2`
  - Efforts of membrane according to `x_{ref}` and `y_{ref}`.
- **Components:** `f3`
  - Normal effort according to `z_{ref}`.
- **Components:** `mf1, mf2`
  - Bending moments of axis `X` and `Y`.

◊ **PLAN = 'E-MAIL'/'MOY'/ 'INF'/ 'SUP'**

Definition of the plan to write the torque of efforts:
- `'MOY'`: one applies the torque of efforts to the excentré average layer;
- `'INF'`: one applies the torque of efforts to the lower skin;
- `'SUP'`: one applies the torque of efforts to the higher skin;
- `'E-MAIL'`: one applies the torque of efforts to the level of the plan of the grid.
9 Other loadings

9.1 Keyword LIAISON_INTERF

LIAISON_INTERF = F ( ♦ MACR_ELEM_DYNA = macrel, [macr_elem_dyna]
               ◊ TYPE_LIAISON = /'RIGID', [DEFECT]
               /'FLEXIBLE',
               )

The keyword factor LIAISON_INTERF allows to define linear relations between the physical degrees of freedom of the interfaces of the part of model in finite elements and the generalized coordinates of modes of reduced representation of the movements of interface contained in certain macronutrients of static condensation. It is usable with a model containing at the same time finite elements and static macronutrients condensing certain under-fields.

♦ MACR_ELEM_DYNA

Name of macr_elem_dyna who is used to define the linear relations between the physical degrees of freedom of the interface between the field non-condensed modelled in finite elements and a field condensed by the macronutrient and the components of the node compared to generalized coordinates of modes of movements of interface. That is necessary only when the modes of movements of interface are a reduced base of all the constrained modes corresponding each one to a mode of displacement for each degree of physical freedom of the interface.

Relations of the type thus are generated LIAISON_DDL whose coefficients are calculated in a transparent way for the user between the nodes of the dynamic interface of the macronutrient and those associated with the base of reduction which was used to constitute the macronutrient.

◊ TYPE_LIAISON = /'RIGID'//'FLEXIBLE'

• If /'RIGID/', one writes the relation between the physical degrees of freedom of the interface $U_X$ and components of the node compared to generalized coordinates $q$ modes of movements of interface $\Phi$ in the simple shape of product: $U_X=\Phi q$. This choice makes it possible to have a connection more rigid than by taking into account all the constrained modes corresponding each one to a mode of displacement for each degree of physical freedom of the interface.

• If /'FLEXIBLE/', one writes the relation between the physical degrees of freedom of the interface $U_X$ and components of the node compared to generalized coordinates $q$ modes of movements of interface $\Phi$ in the double shape of product: $\Phi^T U_X=\Phi^T \Phi q$. This choice makes it possible to have a connection more flexible than by taking into account all the constrained modes corresponding each one to a mode of displacement for each degree of physical freedom of the interface.

9.2 Keyword RELA_CINE_BP

RELA_CINE_BP = F ( ♦ CABLE_BP = cabl_pr, [cabl_precont]
               ◊ SIGM_BPEL = /'NOT', [DEFECT]
               /'YES',
               ◊ RELA_CINE = /'YES', [DEFECT]
               /'NOT',
               ◊ DIST_MIN = dmin, [R]
               ◊ TYPE_EPX = /'ADHE', [DEFECT]
               /'GLIS',
               /'FROT',
               )

This kind of loading can be defined for a mechanical system including a structure concrete and its cables of prestressing. Initial profiles of tension in the cables, as well as the coefficients of the relations kinematics between the degrees of freedom of the nodes of the cables and the degrees of
freedom nodes of the structure concrete are beforehand given by the operator DEF/ CABLE_BP [U4.42.04]. Concepts cabl_precont produced by this operator all the necessary information bring to the definition of the loading.

The multiple occurrences are authorized for the keyword factor RELA_CINE_BP, in order to allow in the same call to the operator AFFE_CHAR_MECA to define the contributions of each group of cables having been the object of distinct calls to the operator DEF/ CABLE_BP [U4.42.04]. With each group of cables considered, defined by a concept cabl_precont, an occurrence with the keyword factor is associated RELA_CINE_BP.

The loading thus defined is then used to calculate the state of balance of the unit structure concrete/cables of prestressing. However, the taking into account of this kind of loading is not effective in all the operators of resolution. The loading of the type RELA_CINE_BP is recognized for the moment only by the operator STAT_NON_LINE [U4.51.03], incrémentaux behaviors exclusively.

♦ CABLE_BP
Concept of the type cabl_precont product by the operator DEF/ CABLE_BP [U4.42.04]. This concept brings on the one hand the map of the initial constraints in the elements of the cables of the same group, and on the other hand the lists of the relations kinematics between the degrees of freedom of the nodes of these cables and the degrees of freedom of the nodes of the structure concrete.

◊ SIGM_BPEL = ‘YES’/’NOT’
Indicator of type text by which one specifies the taking into account of the initial constraints in the cables; the value by default is ‘NOT’.

In the case ‘NOT’, only the liaisonnement kinematic one is taken into account. It is useful if one connects STAT_NON_LINE whereas one has cables of prestressing. For the first STAT_NON_LINE it is necessary to have put ‘YES’, so that one sets up the tension in the cables. On the other hand, for STAT_NON_LINE following, one should regard as loading only the connections kinematics and thus define the loading with SIGM_BPEL = ‘NOT’, if not the tension is counted twice.

Since the restitution the macro one to put in tension the cables, the user should not need any more to do one AFFE_CHAR_MECA with SIGM_BPEL = ‘YES’, that should thus avoid the risks of error.

◊ RELA_CINE = ‘YES’/’NOT’
Indicator of type text by which one specifies the taking into account of the relations kinematics between the degrees of freedom of the nodes of the cables and the degrees of freedom of the nodes of the structure concrete; value by default ‘YES’.

◊ DIST_MIN = dmin
See LI/ AISON_SOLIDE §5.16.

◊ TYPE_EPX = ‘ADHE’/’GLIS’/’FROT’
This keyword has effect only in CALC/EUROPLEXUS. It makes it possible to indicate if one wishes total connections cable-concrete (i.e. in the 3 directions of space, corresponding to this loading Aster), of the sliding joints or rubbing.

9.3 Keyword IMPE_FACE

IMPE_ FACE = _F ( ♦ GROUP_ MA = lgma, [l_gr_maille]
♦ IMPE = Q, [R] or [function] )

The keyword factor IMPE_ FACE allows to apply an acoustic impedance in the case of modeling 3D_FLUIDE.

♦ Topological assignment: GROUP_ MY
The loading is affected on these meshs which are necessarily triangles or of the quadrangles.

♦ IMPE
Value of the acoustic impedance applied to the face.
9.4 **Keyword VITE_FACE**

VITE_FACE = _F (  
  GROUP_MA = lgma,  
  VNOR = V ,  
)  

The keyword factor VITE_FACE allows to apply normal speeds to a face in the case of modeling 3D_FLUIDE.

- **Topological assignment:** GROUP_MY  
  The loading is affected on these meshes which are necessarily triangles or of the quadrangles.
- **VNOR**  
  Value the normal speed applied to the face.

9.5 **Keyword ONDE_PLANE**

ONDE_PLANE = _F (  
  GROUP_MA = lgma,  
  TYPE_ONDE = ty,  
  DIRECTION = (K_x, K_there, K_z),  
  FONC_SIGNAL = F,  
  DIST = /dist  
  DIST_REFLECHI = dist_r,  
  DEPL_IMPO = fd,  
)  

The keyword factor ONDE_PLANE (only one occurrence is allowed) allows to impose a seismic loading by plane wave, corresponding to the loadings classically met at the time as of calculations of interaction ground-structure by the integral equations (see [R4.05.01]). In harmonic, a wave planes elastic is characterized by its direction, its pulsation and its type (wave P for the compression waves, waves S, SV or SH for the waves of shearing). In transient, the data of the pulsation, corresponding to a standing wave in time, must be replaced by the data of a profile of displacement which one will take into account the propagation in the course of time in the direction of the wave.

More precisely, one characterizes:

- A wave P by the function \( u(x, t) = f(k \cdot x - C_p t) \);
- A wave S by the function \( u(x, t) = f(k \cdot x - C_s t) \wedge k \);

With:

- \( k \) the unit vector of direction;
- \( f \) profile of the wave given according to the direction \( k \).

There can be only one occurrence of this keyword in AFFE_CHAR_MECA.

- **Topological assignment:** GROUP_MY
  The loading is affected on the meshes absorbing borders concerned with the introduction of the incidental wave. If nothing is given, by defaults, they are all the meshes of modeling ABSO who are concerned.
- **DIST**
  This real parameter makes it possible to on the way determine dephasing in time due of the wave, by indicating the entrance point of the loading by plane wave in the structure given by the scalar product of the vector of wave and the position of the point source of entry \( k \cdot x_0 \). If this parameter is not present, then the operand will be informed FONC_SIGNAL with a tablecloth of temporal signals.
- **DIST_REFLECHI**
  This real parameter allows the taking into account of the reflected wave, activated if one gives the position of the exit point \( k \cdot x_1 \). In this case, the expression of the profile of the wave by taking account of space dephasing dependent on the way of the wave \( f(k \cdot (x - x_0) - C_m t) \) becomes
  \[
  f(k \cdot (x - x_0) - C_m t) + f(k \cdot (2x_1 - x_0 - x) - C_m t) \quad \text{with} \quad m=S \text{ or } P.
  \]
- **TYPE_ONDE = 'P'/'S'/'SV'/'HS'**

警告：此翻译过程使用的是“机器翻译”。它可能不精确且不准确，且仅供参考。
Type of the wave (compression or shearing).

- $P$ : compression wave
- $SV$ : wave of shearing (only in 3D)
- $SH$ : wave of shearing (only in 3D)
- $S$ : wave of shearing (only in 2D)

Directions of the waves $P$, $SV$, and $SH$ are given starting from the vector $V$ informed by DIRECTION. Namely:

- $P$ is colinear with $V$ and normalized to 1,
- $SH$ is the intersection of the horizontal plane and normal plane with $V$, and normalized to 1,
- $SV$ is the vector product of $SH$ and of $P$. There exists a case of indetermination with this rule when the horizontal plane and the normal plane are confused. In this case, if $V = Z$ purely vertical, one imposes $SH = Y$, and $SV = X$.

♦ DIRECTION
Direction of propagation of the wave.

♦ FONC_SIGNAL
Derived from the profile of the wave $f(t)$ for $t \in [0, +\infty[$. 

Caution: it is the function corresponding at the speed $v(t) = \dot{u}(t)$ that the user gives in FONC_SIGNAL. If the operand DIST is absent, one gives a tablecloth of temporal signals of speed parameterized by values of coordinates of propagation $k \cdot x$ for the variable $X$.

◊ DEPL_IMPO
Profile of the wave $fd(t)$ for $t \in [0, +\infty[$. 

Caution: it is the function corresponding to the integral speed $v(t) = \dot{u}(t)$ that the user gives obligatorily in DEPL_IMPO. This data optional is activated only if there are stiffnesses added on the absorbing border, activated by the data of a nonworthless value of the operand LONG_CARA keyword ELAS of DEFI_MATERIAU. If the operand DIST is absent, one gives a tablecloth of temporal signals in displacement parameterized by values of coordinates of propagation $k \cdot x$ for the variable $X$.

Notice concerning the use of this load:

It is strongly advised to direct by means of the keywords ORIE_PEAU_2D or ORIE_PEAU_3D of MODI_MAILLAGE meshes of skin affected by a modeling of border absorbing towards the outside of the field which they delimit.

### 9.6 Keyword ONDE_FLUI

```
ONDE_FLUI = F ( ♦ GROUP_MA = lgma, [l_gr_maille]
               ♦ NEAR = p , [R] )
```

The keyword factor ONDE_FLUI allows to apply an amplitude of pressure of sinusoidal incidental wave arriving normally at a face to modelings 3D_FLUIDE, 2D_FLUIDE and AXIS_FLUIDE.

♦ Topological assignment: GROUP_MY
The wave is applied to these meshes. They are necessarily meshes of edge (segments in 2D and triangles/quadrangles in 3D).

♦ NEAR
Amplitude of pressure of sinusoidal incidental wave arriving normally at the face.

### 9.7 Keyword FLUX_THM_REP

```
FLUX_THM_REP = F ( ♦ / ALL = ’YES’,
                   / GROUP_MA = lgma, [l_gr_maille]
                   ♦ | FLUN = phiT, [R] or [function]
                   | FLUN_HYDR1 = phiE, [R] or [function] )
```

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 2020 EDF R&D - Licensed under the terms of the GNU FDL (http://www.gnu.org/copyleft/fdl.html)
The keyword factor \texttt{FLUX\_THM\_REF} allows to apply to a field of continuous medium 2D or 3D a heat flow and/or a contribution of fluid mass (hydraulic flow). Hydraulic flows (water and vapor) are defined by:

\begin{equation}
\begin{aligned}
    \phi_e^c &= \rho_c \left( \nabla P_e - \rho_c g \right) \cdot n \\
    \phi_e^v &= \rho_v \left( \nabla P_v - \rho_v g \right) \cdot n
\end{aligned}
\end{equation}

With the density of water $\rho_c$, density of the vapor $\rho_v$, pressure of water $P_e$ (degree of freedom \texttt{PRE1}) and the steam pressure $P_v$ (degree of freedom \texttt{PRE2}).

The heat flow is defined by:

\begin{equation}
\phi_T = \lambda_T \frac{\partial T}{\partial n} + h_m \phi_e^c + h_m \phi_v^v + h_m \phi_a^a
\end{equation}

With the mass enthalpy of water $h_m^l$, vapor $h_m^v$, air $h_m^a$ and the flow of air $\phi_a^a$.

- **Topological assignment**: \texttt{ALL}, \texttt{GROUP\_MY}
  - The loading is affected on these meshes They are necessarily meshes of edge (segments in 2D and triangles/quadrangles in 3D). This loading is usable only in (thermo) - hydraulic or hydraulics pure (\texttt{THM}, \texttt{THH}, \texttt{THHM}, \texttt{HM} or \texttt{HHM}).
- \texttt{FLUN}
  - Value of the heat flow.
- \texttt{FLUN\_HYDR1}
  - Value of the hydraulic flow associated with the component water.
- \texttt{FLUN\_HYDR2}
  - Value of the hydraulic flow associated with the component vapor.
- \texttt{FLUN\_TAILCOAT}
  - Value of hydraulic flow injected in a cohesive interface for models HM-XFEM. This loading is imposed exclusively using a function of space on the principal elements and not the elements of edge. For more details concerning the use of this keyword, to refer to documentations [R7.02.18] and [U2.05.02].

### 9.8 **Keyword \texttt{FORCE\_SOL}**

\begin{verbatim}
\end{verbatim}

The keyword factor \texttt{FORCE\_SOL} allows to take into account the force intern of a field of ground by using the temporal evolutions of the contributions in rigidity, mass and damping of the impedance of ground. The impedance of ground extracted at the initial moment makes it possible to constitute by \texttt{MACR\_ELEM\_DYNA} a macronutrient representing the behavior of the field of ground which one adds to the model of structure. The dynamic interface of the macronutrient is described either by a super-mesh of the model containing at the same time the structure and this macronutrient, or by a group of nodes if the physical interface coincides with the modal dynamic interface.

One can also take into account, if it exists, the temporal evolution of the seismic forces, assigned to this same dynamic interface in the form of logical unit.
This kind of load is taken into account in the order `DYNA_NON_LINE`. An example of use is provided in the case `youStMISS03B [V1.10.122]`. There can be only one occurrence of this keyword in `AFFE_CHAR_MECA`.

- **Topological assignment:** `SUPER_MAILLE/GROUP_NO_INTERF`

These operands make it possible to describe the dynamic interface DU macronutrient representing the behavior of the field of ground which one adds to the model of structure is by a super-mesh of the model containing at the same time the structure and this macronutrient by the keyword `SUPER_MAILLE`, that is to say by a group of nodes by the keyword `GROUP_NO_INTERF` if the physical interface coincides with the modal dynamic interface.

- **UNITE_RESU_RIGI/UNITE_RESU_AMOR/UNITE_RESU_MASS**

These operands make it possible to introduce temporal evolutions of the contributions in rigidity, mass and damping of the impedance of ground in the form of logical units.

- **Operand `TYPE`**

This operand makes it possible to introduce the common format of writing (‘ASCII’ or ‘BINAIRE’) temporal evolutions of the contributions in rigidity, mass and damping of the impedance of ground informed in the form of logical units by the operands precedents `UNITE_RESU_*`. The value of the operand must be coherent with the value entered the operand `TYPE_FICHIER_TEMPS` during the calculation of the option `TYPE_RESU='FICHIER_TEMPS'` by `CALC_MISS`.

- **Operand `UNITE_RESU_FORC`**

This operand makes it possible to introduce, if it exists and in the form of logical unit, temporal evolution of the seismic forces, assigned to the dynamic interface of the macronutrient representing the behavior of the field of ground which one adds to the model of structure.

- **Operand `NB_PAS_TRONCATURE`**

This operand makes it possible to introduce (if it is present) a truncation of the calculation of the contribution to the second member of the force of ground into `DYNA_NON_LINE`. This truncation takes place on the summation starting from the fields of displacements stored at the previous moments. Following tests, one realizes that it is often enough to truncate the sum on the 50 to the last 100 steps of time instead of making the complete sum over every preceding stored moment, which is the option by default. It is thus advised to use as value practise the whole value of the maximum between 50 and the tenth of the full number of step of stored times.