

Operators AFFE_CHAR_CINE and AFFE_CHAR_CINE_F

1 Goal

To define a loading of the type “imposed degrees of freedom”.

This order can be used with a mechanical, thermal or acoustic model. The treatment of these conditions “kinematics” will be done without dualisation and thus without addition of degrees of freedom of Lagrange.

- 1) For AFFE_CHAR_CINE (except for the case EVOL_IMPO) , the affected values do not depend on any parameter and are defined by actual values (mechanics or thermics) or complex values (acoustics). These values can be worthless (blocking).
- 2) For AFFE_CHAR_CINE_F, the affected values are functions of one (or several) parameters to be chosen as a whole (*INST* , *X* , *Y* , *Z*) .

Product a structure of data of the type `char_cine_*` .

2 General syntax

```
CH [char_cine_*] = AFFE_CHAR_CINE

    ( ♦ MODEL = Mo , [model]

      ♦ / MECA_IMPO = (see keyword MECA_IMPO),
        / THER_IMPO = (see keyword THER_IMPO),
        / ACOU_IMPO = (see keyword ACOU_IMPO),
        / EVOL_IMPO = evoimp / [evol_ther]
                               / [evol_elas]
                               / [evol_noli]

      ◊ NOM_CMP = lcmp [l_TXM]

    )

if MECA_IMPO then [*] meca
if THER_IMPO then ther
if ACOU_IMPO then acou
if EVOL_IMPO then [*] meca or ther (according to
evoimp)
```

```
CH [char_cine_*] = AFFE_CHAR_CINE_F

    ( ♦ MODEL = Mo , [model]

      ♦ / MECA_IMPO = (see keyword MECA_IMPO),
        / THER_IMPO = (see keyword THER_IMPO),

    )

if MECA_IMPO then [*] meca
if THER_IMPO then ther
```

3 General information

These two orders create concepts of the type `char_cine_*` (`_meca/_ther`).

The order `AFFE_CHAR_CINE` can also create concepts of the type `char_cine_acou`.

These types are different from the type `load` created by the orders `AFFE_CHAR_MECA` [U4.44.01], `AFFE_CHAR_THER` [U4.44.02] or `AFFE_CHAR_ACOU` [U4.44.04]).

The objects created are thus not interchangeable.

The advantage of the loads “kinematics” is that they do not increase the number of unknown factors of the systems to be solved, contrary to the method of dualisation by multipliers of LAGRANGE, used in the orders producing a concept of the type `load`.

On the other hand, the use of these loads comprises the following limitations:

- one can use them only in the case of relation of the type “ddl imposed” (and not for linear relations),
- these loads are not yet allowed in all the total orders. Today the possible orders are:
 - `MECA_STATICS`, `STAT_NON_LINE`, `DYNA_NON_LINE`
 - `THER_LINEAIRE`
- for a calculation not using the total orders: assembly of a matrix, then resolution, the sequence of orders to be used is more complicated than with “ordinary” loads as one can see it in example 2 [§ 5.2].

4 Operands

4.1 General information on the operands

Operands under the keywords factors MECA_IMPO , THER_IMPO and ACOU_IMPO are of two forms:

- operands specifying the geometrical entities on which the loadings (keywords are affected GROUP_MA , GROUP_NO , NODE ...). The arguments of these operands are identical for the two operators.
- operands specifying the affected values (DX , DY , DZ , etc...). The significance of these operands is the same one for the two operators. The arguments of these operands are all of the real type for the operator AFFE_CHAR_CINE and of the type function (or formula) for the operator AFFE_CHAR_CINE_F .

This is true near with an exception: the keyword factor ACOU_IMPO (which does not exist in the order AFFE_CHAR_CINE_F) is always of complex type.

We will thus not distinguish in this document, except fast mention of the opposite, the two operators AFFE_CHAR_CINE and AFFE_CHAR_CINE_F .

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

- 1) maybe by the operand ALL = 'YES' who allows to indicate all the nodes of the grid,
- 2) maybe by the operand GROUP_NO allowing to indicate a list of groups of nodes,
- 3) maybe by the operand NODE allowing to indicate a list of nodes.
- 4) maybe by the operands GROUP_MA and MESH allowing to indicate all the nodes carried by the meshes indicated by the lists of MESH and of GROUP_MA .

4.2 Behavior in the event of overload:

4.2.1 Overload within one only order AFFE_CHAR_CINE

When one uses within the same order, several occurrences of MECA_IMPO (or THER_IMPO ,...) and that certain nodes are affected several times, it is the last occurrence which precedes. For example:

```
chcine= AFFE_CHAR_CINE (MECA_IMPO= (
      _F (TOUT=' OUI',      DX= 1. ,...)
      _F (NOEUD=' N3',      DX= 3. ,...)
```

In this case, imposed displacement DX for the node N3 is worth: 3.

4.2.2 Overload between several orders AFFE_CHAR_CINE

If several different orders are used, the behavior is different. For example:

```
chcin1= AFFE_CHAR_CINE (MECA_IMPO= _F (TOUT=' OUI',      DX= 1. ,...)
chcin2= AFFE_CHAR_CINE (MECA_IMPO= _F (NOEUD=' N3',      DX= 3. ,...)
```

In this case, imposed displacement DX for the node N3 is worth: 4 (because 1+3)

4.2.3 Overload enters AFFE_CHAR_CINE and AFFE_CHAR_MECA

If one "mixes" the orders AFFE_CHAR_MECA and AFFE_CHAR_CINE , the code will stop in fatal error (FACTOR_41) by explaining that there is a superabundant relation of blocking (NODE N3 / DX).

4.3 Operand MODEL

◆ MODEL = Mo

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

4.4 Keyword MECA_IMPO

4.4.1 Goal

Keyword factor usable to impose, with nodes or groups of nodes, a value of displacement, definite component by component in the total reference mark.

These boundary conditions will be treated, thereafter, by the method known as of elimination of the imposed degrees of freedom (i.e. without dualisation, contrary to the treatment of the same type of limiting condition by the use of the operators AFFE_CHAR_MECA or AFFE_CHAR_MECA_F [U4.44.01]).

4.4.2 Syntax

AFFE_CHAR_CINE

```
 / MECA_IMPO = ( _F ( ◆ / ALL = 'YES' ,
 / | NODE = lno , [l_noeud]
 | GROUP_NO = lgno , [l_gr_noeud]
 | MESH = lma , [l_maille]
 | GROUP_MA = lgma , [l_gr_maille]
 ◆ | DX = ux , [R]
 | DY = uy , [R]
 | ... (see the complete listing below)
 ) , ) ,
```

AFFE_CHAR_CINE_F

```
 / MECA_IMPO = ( _F ( ◆ / ALL = 'YES' ,
 / | NODE = lno , [l_noeud]
 | GROUP_NO = lgno , [l_gr_noeud]
 | MESH = lma , [l_maille]
 | GROUP_MA = lgma , [l_gr_maille]
 ◆ | DX = U_xf , [function
 ( * ) ]
 | DY = U_yf , [function
 ( * ) ]
 | ... (see the complete listing below)
 ) , ) ,
```

function (*) : function or formula

List of the keywords available under MECA_IMPO in AFFE_CHAR_CINE :

'DRX'	'DRY MARTINI'	'DRZ'	'DX'	'DY'	'DZ'
'E1X'	'E1Y'	'E1Z'	'E2X'	'E2Y'	'E2Z'
'E3X'	'E3Y'	'E3Z'	'E4X'	'E4Y'	'E4Z'
'GONF'	'GRX'	'H1X'	'H1Y'	'H1Z'	'PHI'
'PRE1'	'PRE2'	'CLOSE'	'PRES11'	'PRES12'	'PRES13'
'PRES21'	'PRES22'	'PRES23'	'PRES31'	'PRES32'	'PRES33'
'TEMP'	'UI2'	'UI3'	'UI4'	'UI5'	'UI6'
'UO2'	'UO3'	'UO4'	'UO5'	'UO6'	'V11'
'V12'	'V13'	'V21'	'V22'	'V23'	'V31'
'V32'	'V33'	'VI2'	'VI3'	'VI4'	'VI5'
'VI6'	'VO2'	'VO3'	'VO4'	'VO5'	'VO6'
'WI1'	'WI2'	'WI3'	'WI4'	'WI5'	'WI6'
'WO'	'WO1'	'WO2'	'WO3'	'WO4'	'WO5'
'WO6'	'LH1'	'GLIS'			

List of the keywords available under MECA_IMPO in AFFE_CHAR_CINE_F :

'DRX'	'DRY MARTINI'	'DRZ'	'DX'	'DY'	'DZ'
'GRX'	'PRE1'	'PRE2'	'CLOSE'	'TEMP'	'PHI'
'GLIS'					

They are the names of the degrees of freedom carried by the finite elements of the model. The significance of these names is to be sought in the documentation of the finite elements.

4.4.3 Operands

/ MECA_IMPO

DX = ux or ux _f	Value of the component of displacement
DY = uy or uy _f	in translation imposed
DZ = uz or uz _f	on the specified nodes

Only for the nodes of a model 3D comprising of the elements of beam, plate, hull, discrete:

DRX = drx or drx _f	Value of the component of displacement
DRY MARTINI = dry	in rotation imposed
Martini or dry _f	on the specified nodes
DRZ = drz or drz _f	

For the "exotic" degrees of freedom more: GRX , TEMP , NEAR and PHI , one will refer to the documentation of the order AFFE_CHAR_MECA [U4.44.01 §3.9].

Caution:

It is checked that the degree of freedom specified exists in this node for at least one of the elements of the model (keyword MODEL) who are based on this node.

Moreover, the rule of overload is observed when the same degree of freedom of the same node is imposed several times: only the last value is retained.

4.5 Keyword THER_IMPO

4.5.1 Goal

Keyword factor usable to impose, with nodes or groups of nodes, a value of nodal temperature.

These boundary conditions will be treated, thereafter, by the method known as of elimination of the imposed degrees of freedom (i.e.: without dualisation contrary to the treatment of the same type of condition limits by the use of the operators AFFE_CHAR_THER or AFFE_CHAR_THER_F [U4.44.02])

4.5.2 Syntax

for AFFE_CHAR_CINE

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

for AFFE_CHAR_CINE_F

```
/ THER_IMPO = ( _F ( ♦ / ALL = 'YES' ,  
/ | NODE = lno , [l_noeud]  
| GROUP_NO = lgno , [l_gr_noeud]  
| MESH = lma , [l_maille]  
| GROUP_MA = lgma , [l_gr_maille]  
♦ | TEMP = ft , [function ( * )]  
| TEMP_SUP = ftsup , [function ( * )]  
| TEMP_INF = ftinf , [function ( * )]  
) , ) ,
```

function (*) : function or formula

4.5.3 Operands

- | TEMP
Temperature imposed on the nodes (or on the average layer for the thermal hulls)
- | TEMP_INF
Temperature imposed on the lower face for the thermal elements of hulls.
- | TEMP_SUP
Temperature imposed on the higher face for the thermal elements of hulls.

For the hulls, the faces lower and higher are defined, mesh by mesh, the direction of the normal external deduced from classification of the nodes: to see FACE_IMPO of AFFE_CHAR_MECA [U4.44.01].

4.6 Keyword ACOU_IMPO

4.6.1 Goal

Keyword factor usable to impose, with nodes or groups of nodes, a value of acoustic pressure.

These boundary conditions will be treated, thereafter, by the method known as of elimination of the imposed degrees of freedom (i.e.: without dualisation contrary to the treatment of the same type of condition limits by the use of the operator AFFE_CHAR_ACOU [U4.44.04]).

4.6.2 Syntax

For AFFE_CHAR_CINE

```
/ ACOU_IMPO = ( _F ( ♦ / ALL = 'YES' ,  
/ | NODE = lno , [l_noeud]  
| GROUP_NO = lgno , [l_gr_noeud]  
| MESH = lma , [l_maille]  
| GROUP_MA = lgma , [l_gr_maille]  
♦ CLOSE = p , [C]  
) , ) ,
```

For AFFE_CHAR_CINE_F :

Pas de keyword ACOU_IMPO because there is no yet complex function.

4.6.3 Operands

NEAR

Value of the acoustic pressure complexes imposed on (S) the node (S) specified (S).

4.7 Keyword EVOL_IMPO = evoimp NOM_CMP = ('DX', 'DY')

This keyword allows the “structural zoom” (see for example the test zzzz230a).

The effect of this keyword is to impose all ddls of the evolution evoimp as if they were functions of time. This opportunity is given for the structures of data evol_elas, evol_noli and evol_ther.

To make a “structural zoom”, one should force the ddls only on the nodes of the edge of model “the zoom”. That wants to say that it is in general necessary to project “coarse” calculation on the meshes of the edge of model “the zoom”.

If, moreover, one does not want to impose all the components, the keyword should be used NOM_CMP to choose the components to impose (by default: all).

Note:

- Attention not to use several EVOL_IMPO on common zones (if not there will be office plurality of the specified values)
- Attention with the use of FONC_MULT with EVOL_IMPO : the result will not be can be not until one waits!
- EVOL_IMPO will be used for any value of time understood enters tmin and tmax (extreme values of the moments of the transient EVOL_IMPO). Apart from this interval, one emits a fatal error (prohibited extrapolation).
- If it EVOL_IMPO only one moment has, one allows the “constant” prolongation and one emits an alarm.

5 Examples

5.1 Degrees of freedom imposed in mechanics

```
chcine = AFFE_CHAR_CINE (MODEL = Mo,  
    MECA_IMPO= (  
        _F (ALL = 'YES' , DRZ = 0.),  
        _F (GROUP_NO = 'bord1', DX = 0. , DY = 0. , DZ =  
0. ,  
0. ,)))  
        DRX = 0. , DRY MARTINI =
```

For this problem of plate in the plan XY , one blocks all the degrees of freedom of rotation around Z and one embeds the plate on his edge *bord1* .

5.2 Compared use of the loads kinematics and “ordinary”

5.2.1 Total orders

```
ch1 = AFFE_CHAR_THER (...)  
ch2 = AFFE_CHAR_CINE_F (TEMP_IMPO = _F (...))  
evoth = THER_LINEAIRE ( ...  
    EXCIT = ( _F (LOAD = ch1),  
    _F (LOAD = ch2),)  
    ...)
```

There is no difference.

5.2.2 Calculation “step by step”

Ordinary loads

```
ch1 = AFFE_CHAR_MECA (...)  
mel = CALC_MATR_ELEM (... OPTION = 'RIGI_MECA' , LOAD = ch1)  
subdued = ASSE_MATRICE ( MATR_ELEM = mel...)  
subdued = TO FACTORIZE ( reuse = subdued, MATR_ASSE = subdued)  
U = TO SOLVE ( MATR = subdued, CHAM_NO = F)
```

Loads kinematics

```
ch1 = AFFE_CHAR_CINE (...)  
mel = CALC_MATR_ELEM (... OPTION = 'RIGI_MECA')  
subdued = ASSE_MATRICE ( MATR_ELEM = mel,..., CHAR_CINE = ch1)  
subdued = TO FACTORIZE ( reuse = subdued, MATR_ASSE = subdued,)  
vcine = CALC_CHAR_CINE (... , CHAR_CINE = ch2,)  
U = TO SOLVE ( MATR = subdued, CHAM_NO = F,  
    CHAM_CINE = vcine)
```

The terms induced by the loads kinematics are deferred to the second member what requires the calculation of an additional field to the nodes *vcine* by the order `CALC_CHAR_CINE` [U4.61.03].