Operators AFFE_CHAR_CINE and AFFE_CHAR_CINE_F

1 Drank

To define a loading of the type "imposed degrees of freedom".

This command can be used with a mechanical, thermal or acoustic model. The processing of these "kinematical" conditions will be done without dualisation and thus without addition of degrees of freedom of Lagrange.

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

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

2 General syntax

```
CH [char_cine_*] = AFFE_CHAR_CINE

            (  ♦MODELE    = Mo  ,                              [model]

               ♦/MECA_IMPO      =            (see key word MECA_IMPO),
                  /THER_IMPO    =            (see key word THER_IMPO),
                  /ACOU_IMPO    =            (see key word ACOU_IMPO),
                  /EVOL_IMPO    = evoimp/          [evol_ther]
                                            /   [evol_elas]
                                            /   [evol_noli]
                     ◊NOM_CMP   = lcmp                         [l_TXM]
            )

siMECA_IMPOalors                        [*]          meca
siTHER_IMPOther
siACOU_IMPOacou
siEVOL_IMPOalors                        [*]          meca or ther
(according to evoimp)


CH [char_cine_*] = AFFE_CHAR_CINE_F

            (  ♦MODELE    = Mo  ,                              [model]

               ♦/MECA_IMPO      =            (see key word MECA_IMPO),
                  /THER_IMPO    =            (see key word THER_IMPO),

            )


siMECA_IMPOalors                        [*]       meca
siTHER_IMPOther
```

3 General information

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

Command `AFFE_CHAR_CINE` can also create concepts of the char_cine_acou `type` .

These types are different from the type `charges` created by commands `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 "kinematical" loads is that they do not increase the number of unknowns of the systems to be solved, contrary to the method of dualisation by LAGRANGE multipliers, used in the commands producing a concept of the type `charges` .

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 "d.o.f. imposed" (and not for linear relations),
- these loads are not yet allowed in all the global commands. Today the possible commands are:
  - `MECA_STATIQUE , STAT_NON_LINE, DYNA_NON_LINE`
  - `THER_LINEAIRE`
- for a computation not using the global commands: assembly of a matrix, then resolution, the sequence of commands 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

the operands under the key keys factors MECA_IMPO , THER_IMPO and ACOU_IMPO are of two forms:

- operands specifying the geometrical entities on which the loadings are affected (key keys GROUP_MA , GROUP_NO , NOEUD …). The arguments of these operands are identical for the two operators.
- operands specifying the affected values (DX , DY , DZ , etc…). The meaning of these operands is the same one for the two operators. The arguments of these operands are all of the real type for operator AFFE_CHAR_CINE and of the standard function (or formulates) for operator AFFE_CHAR_CINE_F .

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

We will thus not distinguish in this document, except fast mention of the opposite, 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) either by the operand TOUT = "OUI' which makes it possible to indicate all the nodes of the mesh,
2) or by operand GROUP_NO allowing to indicate a list of nodes groups,
3) or by the operand NOEUD allowing to indicate one nodes list.
4) maybe by the operands GROUP_MA and MESH allowing to indicate all the nodes carried by meshes indicated by the lists of MESH and GROUP_MA .

4.2 Behavior in the event of overload:

4.2.1 Overload within one only command AFFE_CHAR_CINE

When one uses within the same command, several occurrences of MECA_IMPO (or THER_IMPO ,…) and that some 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, displacement imposed DX for the N3 node is worth: 3.

4.2.2 Overload between several commands AFFE_CHAR_CINE

If several different commands 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, displacement imposed DX for the N3 node is worth: 4 (because 1+3)

4.2.3 Overload between AFFE_CHAR_CINE and AFFE_CHAR_MECA

If one “mixes” commands AFFE_CHAR_MECA and AFFE_CHAR_CINE , the code will stop in fatal error (FACTOR_41) by explaining why there is a superabundant relation of blocking (NOEUD N3/DX).

4.3 MODEL operand

```
♦MODELE    = Mo
```

Product concept by the operator `AFFE_MODELE` [U4.41.01] where definite affected ones on the mesh the are element types finished.

4.4 Key word MECA_IMPO

4.4.1 Drank

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

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 processing of the same type of limiting condition by the use of operators `AFFE_CHAR_MECA` or `AFFE_CHAR_MECA_F` [U4.44.01]).

4.4.2 Syntax

```
AFFE_CHAR_CINE

   /MECA_IMPO    =  (_F   ( ♦/TOUT     =         ' OUI'   ,
                            /  | NOEUD =lno            ,
[l_noeud]
                               | GROUP_NO =lgno    ,          [l_gr_noeud]
                               | NET =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   ( ♦/TOUT     =         ' OUI'   ,
                            /  | NOEUD =lno            ,
[l_noeud]
                               | GROUP_NO =lgno    ,          [l_gr_noeud]
                               | NET =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
```

Lists key keys available under `MECA_IMPO` in `AFFE_CHAR_CINE` :

```
“DRX”     “DRY”     “DRZ”     “DX”      “DY”      “DZ”
“E1X”     “E1Y”     “E1Z”     “E2X”     “E2Y”     “E2Z”
“E3X”     “E3Y”     “E3Z”     “E4X”     “E4Y”     “E4Z”
“GONF”    “GRX”     “H1X”     “H1Y”     “H1Z”     “PHI”
“PRE1”    “PRE2”    “NEAR”    “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”
```

List key keys available under `MECA_IMPO` in `AFFE_CHAR_CINE_F` :

```
“DRX”     “DRY”     “DRZ”     “DX”      “DY”      “DZ”
“GRX”     “PRE1”    “PRE2     “NEAR”    “TEMP”    “PHI”
```

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

4.4.3 Operands

/MECA_IMPO

DX = ux or uxf	Value of the component of displacement
DY = uy or uyf	in **translation** imposed
DZ = uz or uzf	on the nodes Only

specified for the nodes D” a model 3D comprising of the beam elements, plates, shell, discrete:

DRX = drx or drxf	Value of the component of displacement
DRY = dry or dryf	in **rotation** imposed
DRZ = drz or drzf	on the nodes specified

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

Caution:

It is checked that the specified degree of freedom exists in this node for at least one of the elements of the model (key word MODELS) which lean 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 Key word THER_IMPO

4.5.1 Drank

Key word factor usable to impose, with nodes or nodes groups, 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 processing of the same type of condition limits by the use of operators AFFE_CHAR_THER or AFFE_CHAR_THER_F [U4.44.02])

4.5.2 Syntax

for AFFE_CHAR_CINE

```
 /THER_IMPO   = (_F  ( ♦   /TOUT  =         ' OUI'   ,
                          /  | NOEUD =lno            ,           [l_noeud]
                             | GROUP_NO =lgno    ,          [l_gr_noeud]
                             | NET =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  ( ♦/TOUT     =         ' OUI'   ,
                          /  | NOEUD =lno            ,           [l_noeud]
                             | GROUP_NO =lgno    ,          [l_gr_noeud]
                             | NET =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 average for the thermal shells)

| TEMP_INF

Temperature imposed on the lower face for the thermal shell elements.

| TEMP_SUP

Temperature imposed on the upper face for the thermal shell elements.

For the shells, the sides lower and higher are defined, mesh by mesh, the direction of the external norm deduced from the numbers of the nodes: to see FACE_IMPO of AFFE_CHAR_MECA [U4.44.01].

4.6 Key word ACOU_IMPO

4.6.1 Drank

Key word factor usable to impose, with nodes or nodes groups, 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 processing of the same type of condition limits by the use of operator AFFE_CHAR_ACOU [U4.44.04]).

4.6.2 Syntax

For AFFE_CHAR_CINE

```
    /ACOU_IMPO    = (_F  ( ♦    /TOUT  =             ' OUI'   ,
                              /   | NOEUD =lno            ,
[l_noeud]
                                  | GROUP_NO =lgno    ,          [l_gr_noeud]
                                  | NET =lma          ,          [l_maille]
                                  | GROUP_MA =lgma    ,
[l_gr_maille]
                           ♦PRES   =p                 ,          [C]
                     ), ),
```

For AFFE_CHAR_CINE_F :

No key word ACOU_IMPO because it does not have yet a complex function there.

4.6.3 Operands

NEAR

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

4.7 Key word EVOL_IMPO = evoimp NOM_CMP = ("DX", "DY")

This key word allows the "structural zoom" (see for example the test zzzz230a).

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

To make a "structural zoom", it should not be forced the ddls that on the nodes of edge of model "the zoom". That wants to say that it is in general necessary to project the "coarse" computation on meshes of edge of model "the zoom".

If, moreover, one does not want to impose all the components, it is necessary to use key word NOM_CMP to choose the components to be imposed (by default: all).

Note:

- *Attention not to use several EVOL_IMPO on zones common (if not there will be office plurality of the specified values)*
- *Attention to the use of FONC_MULT with EVOL_IMPO : result will not be can be not until one waits!*
- *L'EVOL_IMPO will be used for any value of time understood enters $tmin$ and $tmax$ (extreme values of times of transient EVOL_IMPO). Apart from this interval, one emits a fatal error (prohibited extrapolation).*

- *If* `l'EVOl_IMPO` *has only one time, one allows the “constant” prolongation and one emits an alarm.*

5 Degrees of freedom

5.1 examples imposed in mechanics

```
chcine =AFFE_CHAR_CINE      (= Mo MODELS,
             MECA_IMPO= (
                 _F   (TOUT = "OUI'   , DRZ = 0.),
                 _F   (GROUP_NO = "bord1",    DX =0 .,   DY =0 .,   DZ =0 .,
                                           DRX =0 .,   DRY =0    .,)))
```

For this problem of plate in the plane 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 kinematical loads and "ordinary"

5.2.1 Global commands

```
ch1=AFFE_CHAR_THER             (…)
ch2=AFFE_CHAR_CINE_F              (TEMP_IMPO =_F   (…))
evoth=THER_LINEAIRE            ( …
                              EXCIT = ( _F (CHARGE = ch1),
                                        _F (CHARGE = ch2),)
                              …)
```

There is no difference.

5.2.2 Computation "step by step"

ordinary Loads

```
ch1=AFFE_CHAR_MECA          (…)
mel=CALC_MATR_ELEM          (… OPTION = "RIGI_MECA"   , CHARGE = ch1)
matas=ASSE_MATRICE       ( MATR_ELEM = mel…)
matas=FACTORISER     ( reuse = subdued, MATR_ASSE = subdued)
U=RESOUDRE           ( MATR = subdued,  CHAM_NO = F)
```

Kinematical loads

```
ch1=AFFE_CHAR_CINE          (…)
mel=CALC_MATR_ELEM          (… OPTION = "RIGI_MECA")
matas=ASSE_MATRICE       ( MATR_ELEM = mel,…, CHAR_CINE = ch1)
matas=FACTORISER     ( reuse = subdued, MATR_ASSE = subdued,)
vcine=CALC_CHAR_CINE        (…, CHAR_CINE = ch2,)
U=RESOUDRE           ( MATR = subdued,  CHAM_NO = F,
                       CHAM_CINE = vcine)
```

the terms induced by the kinematical loads are deferred to the second member what requires the computation of an additional field at nodes `by the command` vcine `CALC_CHAR_CINE` [U4.61.03].