Opérateurs AFFE_CHAR_CINE et AFFE_CHAR_CINE_F

1 But

Définir un chargement de type "degrés de liberté imposés".

Cette commande peut être utilisée avec un modèle mécanique, thermique ou acoustique. Le traitement de ces conditions "cinématiques" se fera sans dualisation et donc sans ajout de degrés de liberté de Lagrange.

1) Pour AFFE_CHAR_CINE (sauf pour le cas EVOL_IMPO) , les valeurs affectées ne dépendent d'aucun paramètre et sont définies par des valeurs réelles (mécanique ou thermique) ou des valeurs complexes (acoustique). Ces valeurs peuvent être nulles (blocage).
2) Pour AFFE_CHAR_CINE_F, les valeurs affectées sont des fonctions d'un (ou plusieurs) paramètres à choisir dans l'ensemble $(INST, X, Y, Z)$.

Produit une structure de données de type char_cine_* .

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

2 Syntaxe Générale

```
ch[char_cine_*]  = AFFE_CHAR_CINE

              (  ♦  MODELE = mo  ,                              [modele]

                 ♦  /  MECA_IMPO =            (voir mot clé MECA_IMPO),
                    /  THER_IMPO =            (voir mot clé THER_IMPO),
                    /  ACOU_IMPO =            (voir mot clé ACOU_IMPO),
                    /  EVOL_IMPO = evoimp     /  [evol_ther]
                                              /  [evol_elas]
                                              /  [evol_noli]
                        ◊  NOM_CMP = lcmp                     [l_TXM]
              )

si     MECA_IMPO          alors     [*]       meca
si     THER_IMPO                              ther
si     ACOU_IMPO                              acou
si     EVOL_IMPO          alors     [*]       meca ou ther (selon evoimp)


ch[char_cine_*]  = AFFE_CHAR_CINE_F

              (  ♦  MODELE = mo  ,                              [modele]

                 ♦  /  MECA_IMPO =            (voir mot clé MECA_IMPO),
                    /  THER_IMPO =            (voir mot clé THER_IMPO),

              )


si     MECA_IMPO          alors     [*]    meca
si     THER_IMPO                           ther
```


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

3 Généralités

Ces deux commandes créent des concepts de type char_cine_* (_meca/_ther).

La commande AFFE_CHAR_CINE peut également créer des concepts de type char_cine_acou .

Ces types sont différents du type charge créé par les commandes AFFE_CHAR_MECA [U4.44.01], AFFE_CHAR_THER [U4.44.02] ou AFFE_CHAR_ACOU [U4.44.04]).

Les objets créés ne sont donc pas interchangeables.

L'avantage des charges "cinématiques" est qu'elles n'augmentent pas le nombre d'inconnues des systèmes à résoudre, contrairement à la méthode de dualisation par multiplicateurs de LAGRANGE, utilisée dans les commandes produisant un concept de type charge .

En revanche, l'utilisation de ces charges comporte les limitations suivantes :

- on ne peut les utiliser que dans le cas de relation de type "ddl imposé" (et non pour des relations linéaires),
- ces charges ne sont pas encore admises dans toutes les commandes globales. Aujourd'hui les commandes possibles sont :
  - MECA_ STATIQUE, STAT_NON_LINE, DYNA_NON_LINE
  - THER_LINEAIRE
- pour un calcul n'utilisant pas les commandes globales : assemblage d'une matrice, puis résolution, la séquence de commandes à utiliser est plus compliquée qu'avec des charges "ordinaires" comme on peut le voir dans l'exemple 2 [§ 5.2].

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

4 Opérandes

4.1 Généralités sur les opérandes

Les opérandes sous les mots clés facteurs MECA_IMPO , THER_IMPO et ACOU_IMPO sont de deux formes :

- les opérandes spécifiant les entités géométriques sur lesquelles sont affectées les chargements (mots clés GROUP_MA , GROUP_NO...). Les arguments de ces opérandes sont identiques pour les deux opérateurs.
- les opérandes spécifiant les valeurs affectées (DX , DY , DZ , etc ...). La signification de ces opérandes est la même pour les deux opérateurs. Les arguments de ces opérandes sont tous du type réel pour l'opérateur AFFE_CHAR_CINE et du type fonction (ou formule) pour l'opérateur AFFE_CHAR_CINE_F .

Ceci est vrai à une exception près : le mot clé facteur ACOU_IMPO (qui n'existe pas dans la commande AFFE_CHAR_CINE_F) est toujours de type complexe.

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

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

1) soit par l'opérande TOUT = 'OUI' qui permet de désigner tous les nœuds du maillage,
2) soit par l'opérande GROUP_NO permettant de désigner une liste de groupes de nœuds,
3) soit par l'opérande GROUP_MA permettant de désigner tous les nœuds portés par les mailles désignées par les listes de GROUP_MA .

4.2 Comportement en cas de surcharge :

4.2.1 Surcharge au sein d'une seule commande AFFE_CHAR_CINE

Lorsqu'on utilise au sein d'une même commande, plusieurs occurrences de MECA_IMPO (ou THER_IMPO , ...) et que certains noeuds sont affectés plusieurs fois, c'est la dernière occurrence qui prime. Par exemple :

```
chcine= AFFE_CHAR_CINE( MECA_IMPO=(
                  _F( TOUT='OUI',    DX= 1., ...)
                  _F( GROUP_NO='GN3',    DX= 3., ...)
```

Dans ce cas, le déplacement imposé DX pour le noeud GN3 vaut : 3.

4.2.2 Surcharge entre plusieurs commandes AFFE_CHAR_CINE

Si on utilise plusieurs commandes différentes, le comportement est différent. Par exemple :

```
chcin1= AFFE_CHAR_CINE( MECA_IMPO= _F( TOUT='OUI',    DX= 1., ...)
chcin2= AFFE_CHAR_CINE( MECA_IMPO= _F( GROUP_NO='GN3',    DX= 3., ...)
```

Dans ce cas, le déplacement imposé DX pour le noeud GN3 vaut : 4 (car 1+3)

4.2.3 Surcharge entre AFFE_CHAR_CINE et AFFE_CHAR_MECA

Si on "mixe" les commandes AFFE_CHAR_MECA et AFFE_CHAR_CINE , le code s'arrêtera en erreur fatale (FACTOR_41) en expliquant qu'il y a une relation de blocage sur-abondante (NOEUD N3 / DX).

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

4.3 Opérande MODELE

♦ MODELE = mo

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

4.4 Mot clé MECA_IMPO

4.4.1 But

Mot clé facteur utilisable pour imposer, à des nœuds, une valeur de déplacement, définie composante par composante dans le repère global.

Ces conditions aux limites seront traitées, par la suite, par la méthode dite d'élimination des degrés de liberté imposés (c'est à dire sans dualisation, contrairement au traitement du même type de condition limite par l'utilisation des opérateurs AFFE_CHAR_MECA ou AFFE_CHAR_MECA_F [U4.44.01]).

4.4.2 Syntaxe

```
AFFE_CHAR_CINE

    /   MECA_IMPO =   (_F (   ♦   /   TOUT =        'OUI'  ,
                                  | GROUP_NO = lgno   ,          [l_gr_noeud]
                                  | GROUP_MA = lgma   ,          [l_gr_maille]
                              ♦   |   DX =          ux     ,     [R]
                                  |   DY =          uy     ,     [R]
                                  |   ... (voir la liste complète ci-dessous)
                          ),),

AFFE_CHAR_CINE_F

    /   MECA_IMPO =   (_F (   ♦   /   TOUT =        'OUI'  ,
                                  | GROUP_NO = lgno   ,          [l_gr_noeud]
                                  | GROUP_MA = lgma   ,          [l_gr_maille]
                              ♦   |   DX =          u xf       ,
    [fonction( * )]
                                  |   DY =          u yf       ,
    [fonction( * )]
                                  |   ... (voir la liste complète ci-dessous)

                          ),),

fonction( * ) : fonction ou formule
```


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

Liste des mots clés disponibles sous MECA_IMPO dans AFFE_CHAR_CINE :

DX	UO4	WO3	V22	HHO_U4
DY	UO5	WO4	V23	HHO_U5
DZ	UO6	WO5	V31	HHO_U6
DRX	VI2	WO6	V32	HHO_V1
DRY	VI3	WO	V33	HHO_V2
DRZ	VI4	WI1	PRES11	HHO_V3
GRX	VI5	WO1	PRES12	HHO_V4
PRES	VI6	GONF	PRES13	HHO_V5
PHI	VO2	H1X	PRES21	HHO_V6
TEMP	VO3	H1Y	PRES22	HHO_W1
PRE1	VO4	H1Z	PRES23	HHO_W2
PRE2	VO5	H1PRE1	PRES31	HHO_W3
UI2	VO6	K1	PRES32	HHO_W4
UI3	WI2	K2	PRES33	HHO_W5
UI4	WI3	K3	LH1	HHO_W6
UI5	WI4	V11	GLIS	
UI6	WI5	V12	HHO_U1	
UO2	WI6	V13	HHO_U2	
UO3	WO2	V21	HHO_U3	

Liste des mots clés disponibles sous MECA_IMPO dans AFFE_CHAR_CINE_F :

DX	UI4	VO4	WO1	V32
DY	UI5	VO5	GONF	V33
DZ	UI6	VO6	H1X	PRES11
DRX	UO2	WI2	H1Y	PRES12
DRY	UO3	WI3	H1Z	PRES13
DRZ	UO4	WI4	K1	PRES21
GRX	UO5	WI5	K2	PRES22
PRES	UO6	WI6	K3	PRES23
PHI	VI2	WO2	V11	PRES31
PSI	VI3	WO3	V12	PRES32
TEMP	VI4	WO4	V13	PRES33
PRE1	VI5	WO5	V21	LH1
PRE2	VI6	WO6	V22	GLIS
UI2	VO2	WO	V23	
UI3	VO3	WI1	V31	

Ce sont les noms des degrés de liberté portés par les éléments finis du modèle. La signification de ces noms est à chercher dans la documentation de AFFE_CHAR_MECA [U4.44.01].

4.4.3 Opérandes

/ MECA_IMPO

DX = ux ou uxf
DY = uy ou uyf
DZ = uz ou uzf
Valeur de la composante de déplacement en **translation** imposée sur les nœuds spécifiés

Uniquement pour les nœuds d'un modèle 3D comportant des éléments de poutre, plaque, coque, discret :

DRX = drx ou drxf
DRY = dry ou dryf
DRZ = drz ou drzf
Valeur de la composante de déplacement en **rotation** imposée sur les nœuds spécifiés

Pour les degrés de liberté plus "exotiques" : GRX , TEMP , PRES et PHI , on se reportera à la documentation de la commande AFFE_CHAR_MECA [U4.44.01 §3.9].

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

Attention :

On vérifie que le degré de liberté spécifié existe en ce nœud pour au moins un des éléments du modèle (mot clé `MODELE` *) qui s'appuient sur ce nœud.*

De plus, la règle de surcharge est appliquée quand le même degré de liberté d'un même nœud est imposé plusieurs fois : on ne retient que la dernière valeur.

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

4.5 Mot clé THER_IMPO

4.5.1 But

Mot clé facteur utilisable pour imposer, à des nœuds, une valeur de température nodale.

Ces conditions aux limites seront traitées, par la suite, par la méthode dite d'élimination des degrés de liberté imposés (c'est à dire : sans dualisation contrairement au traitement du même type de condition limite par l'utilisation des opérateurs AFFE_CHAR_THER ou AFFE_CHAR_THER_F [U4.44.02])

4.5.2 Syntaxe

pour AFFE_CHAR_CINE

```
 / THER_IMPO = (_F  (   ♦    / TOUT =       'OUI'  ,
                               | GROUP_NO = lgno   ,          [l_gr_noeud]
                               | GROUP_MA = lgma   ,          [l_gr_maille]
                          ♦   | TEMP     =   t      ,          [R]
                              | TEMP_SUP =   tsup   ,          [R]
                              | TEMP_INF =   tinf   ,          [R]
                  ), ),
```

pour AFFE_CHAR_CINE_F

```
/  THER_IMPO = (_F  (   ♦   /   TOUT =       'OUI'  ,
                               | GROUP_NO = lgno   ,          [l_gr_noeud]
                               | GROUP_MA = lgma   ,          [l_gr_maille]
                          ♦   | TEMP     =   ft     ,          [fonction( * )]
                              | TEMP_SUP =   ftsup  ,          [fonction( * )]
                              | TEMP_INF =   ftinf  ,          [fonction( * )]
                  ), ),
```

```
fonction( * ) : fonction ou formule
```

4.5.3 Opérandes

| TEMP

Température imposée sur les nœuds (ou sur le feuillet moyen pour les coques thermiques)

| TEMP_INF

Température imposée sur la face inférieure pour les éléments de coques thermiques.

| TEMP_SUP

Température imposée sur la face supérieure pour les éléments de coques thermiques.

Pour les coques, les faces inférieure et supérieure sont définies, maille par maille, par la direction de la normale extérieure déduite de la numérotation des nœuds : voir FACE_IMPO de AFFE_CHAR_MECA [U4.44.01].

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

4.6 Mot clé ACOU_IMPO

4.6.1 But

Mot clé facteur utilisable pour imposer, à des nœuds, une valeur de pression acoustique.

Ces conditions aux limites seront traitées, par la suite, par la méthode dite d'élimination des degrés de liberté imposés (c'est à dire : sans dualisation contrairement au traitement du même type de condition limite par l'utilisation de l'opérateur AFFE_CHAR_ACOU [U4.44.04]).

4.6.2 Syntaxe

Pour AFFE_CHAR_CINE

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

Pour AFFE_CHAR_CINE_F :

Pas de mot clé ACOU_IMPO car il n'y a pas encore de fonction complexe.

4.6.3 Opérandes

PRES

Valeur de la pression acoustique complexe imposée sur le(s) nœud(s) spécifié(s).

4.7 Mot clé EVOL_IMPO

```
/   EVOL_IMPO = evoimp      /   [evol_ther]
                            /   [evol_elas]
                            /   [evol_noli]
    ◊    NOM_CMP = lcmp         [l_TXM]
```

Ce mot clé permet le "zoom structural" (voir par exemple le test zzzz230a).

L'effet de ce mot clé est d'imposer **tous** les ddls de l'évolution evoimp comme si c'étaient des fonctions du temps. Cette possibilité est offerte pour les structures de données evol_elas, evol_noli et evol_ther.

Pour faire un « zoom structural », il ne faut imposer les ddls que sur les nœuds du bord du modèle « zoom ». Cela veut dire qu'il faut en général projeter le calcul « grossier » sur les mailles du bord du modèle « zoom ».

Si, de plus, on ne veut pas imposer toutes les composantes, il faut utiliser le mot clé NOM_CMP pour choisir les composantes à imposer (par défaut : toutes).

Remarques :

- *Attention à ne pas utiliser plusieurs EVOL_IMPO sur des zones communes (sinon il y aura cumul des valeurs imposées)*
- *Attention à l'usage de FONC_MULT avec EVOL_IMPO : le résultat ne sera peut être pas ce que l'on attend !*

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

- *L'EVOL_IMPO sera utilisé pour toute valeur du temps comprise entre $tmin$ et $tmax$ (valeurs extrêmes des instants du transitoire EVOL_IMPO). En dehors de cet intervalle, on émet une erreur fatale (extrapolation interdite).*
- *Si l'EVOl_IMPO n'a qu'un instant, on permet le prolongement "constant" et on émet une alarme.*

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

5 Exemples

5.1 Degrés de liberté imposés en mécanique

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

Pour ce problème de plaque dans le plan XY , on bloque tous les degrés de liberté de rotation autour de Z et on encastre la plaque sur son bord $bord1$.

5.2 Utilisation comparée des charges cinématiques et "ordinaires"

5.2.1 Commandes globales

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

Il n'y a pas de différence.

5.2.2 Calcul "pas à pas"

Charges ordinaires

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

Charges cinématiques

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

Les termes induits par les charges cinématiques sont reportés au second membre ce qui nécessite le calcul d'un champ aux nœuds supplémentaire `vcine` par la commande `CALC_CHAR_CINE` [U4.61.03].

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