

## 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_*`.

## 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 then
si ACOU_IMPO then
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 then
```

## 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].

## 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).

## 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

Liste des mots clés disponibles sous MECA\_IMPO dans AFFE\_CHAR\_CINE :

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

Liste des mots clés disponibles sous MECA\_IMPO dans AFFE\_CHAR\_CINE\_F :

DX	UI5	VO5	GONF	V33
DY	UI6	VO6	H1X	PRES11
DZ	UO2	WI2	H1Y	PRES12
DRX	UO3	WI3	H1Z	PRES13
DRY	UO4	WI4	K1	PRES21
DRZ	UO5	WI5	K2	PRES22
GRX	UO6	WI6	K3	PRES23
PRES	VI2	WO2	V11	PRES31
PHI	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	
UI4	VO4	WO1	V32	

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	Valeur de la composante de déplacement
DY = uy ou uyf	en <b>translation</b> imposée
DZ = uz ou uzf	sur les nœuds spécifiés

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

DRX = drx ou drxf	Valeur de la composante de déplacement
DRY = dry ou dryf	en <b>rotation</b> imposée
DRZ = drz ou drzf	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].

**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.

## 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].

## 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 !

- 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.

## 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].