

## Opérateur CREA\_CHAMP

---

### 1 But

---

Créer un champ de type `cham_no`, `carte` ou `cham_elem`. On peut créer un champ

- par affectation de valeurs sur des nœuds ou des mailles,
- en assemblant (et / ou combinant) des morceaux de champs existants,
- en évaluant les fonctions d'un champ de fonctions pour en faire un champ de réels,
- en modifiant la représentation géométrique d'un champ (passage nœuds  $\Leftrightarrow$  points de Gauss par exemple),
- en extrayant un champ d'une structure de données de type `SD_RESULTAT` (`evol_ther`, `evol_noli`, `mode_meca`, ...).
- en extrayant un champ d'une structure de données de type `TABLE`,  
On peut aussi se servir de cette commande pour combiner linéairement plusieurs champs, mais on peut faire des combinaisons plus "compliquées" : multiplication de champs ...

Produit une structure de données de type `cham_no` ou `carte` ou `cham_elem`

## Table des matières

|  |    |
|--|----|
| 1 But.....   | 1  |
| 2 Syntaxe.....   | 4  |
| 3 Opérandes.....   | 8  |
| 3.1 Opérandes généraux .....   | 8  |
| 3.1.1 Opérande TYPE_CHAM, affectation d'un type au champ résultat .....                  | 8  |
| 3.1.2 Opérande MAILLAGE = mailla.....  | 9  |
| 3.1.3 Opérandes MODELE, OPTION, PROL_ZERO.....   | 9  |
| 3.1.4 Mots clé NUME_DDL et CHAM_NO.....  | 9  |
| 3.1.5 Opérande TITRE = titr.....   | 9  |
| 3.1.6 Opérande INFO = /1 /2.....   | 10 |
| 3.2 Opérande OPERATION = 'AFFE' / 'ASSE' / 'EVAL' / 'DISC' / 'EXTR' / 'R2C' / 'C2R'..... | 10 |
| 3.3 Opérandes pour OPERATION = 'AFFE'.....   | 10 |
| 3.3.1 Mot clé facteur AFFE.....  | 10 |
| 3.3.2 Opérandes TOUT='OUI', GROUP_MA, GROUP_NO, MAILLE et NOEUD.....                     | 10 |
| 3.3.3 Opérande NOM_CMP.....  | 10 |
| 3.3.4 Opérandes VALE, VALE_I, VALE_C ou VALE_F.....                                      | 10 |
| 3.3.5 Exemples.....  | 11 |
| 3.4 Opérandes pour OPERATION = 'ASSE'.....   | 11 |
| 3.4.1 Généralités.....   | 11 |
| 3.4.2 Opérandes MAILLAGE, MODELE et PROL_ZERO.....                                       | 11 |
| 3.4.3 Opérandes pour le mot clé facteur ASSE.....  | 11 |
| 3.4.4 Exemples.....  | 12 |
| 3.5 Opérandes pour OPERATION = 'EVAL'.....   | 13 |
| 3.5.1 Opérande CHAM_F= chf.....  | 13 |
| 3.5.2 Opérande CHAM_PARA= l_chpara.....  | 13 |
| 3.5.3 Exemples.....  | 13 |
| 3.6 Opérandes pour OPERATION = 'DISC'.....   | 14 |
| 3.6.1 Opérande CHAM_GD= ch1.....   | 15 |
| 3.6.2 mots clé MODELE et PROL_ZERO.....  | 15 |
| 3.6.3 Exemple.....   | 15 |
| 3.7 Opérandes pour OPERATION = 'NORMALE'.....  | 15 |
| 3.8 Opérandes pour OPERATION = 'EXTR'.....   | 16 |
| 3.8.1 Typage du champ résultat, mot clé TYPE_CHAM.....                                   | 16 |
| 3.8.2 Opérande TABLE.....  | 17 |
| 3.8.3 Opérande RESULTAT.....   | 17 |
| 3.8.4 Opérande NOM_CHAM.....   | 17 |
| 3.8.5 Opérande SENSIBILITE = para.....   | 18 |
| 3.8.6 Opérandes NUME_ORDRE / NUME_MODE / NOM_MODE / ... / INTERPOL.....                  | 18 |

|  |                    |
|--|--------------------|
| <a href="#">3.8.7 Calcul d'un champ contenant les "extrema" d'une SD_RESULTAT.....</a> | <a href="#">18</a> |
| <a href="#">3.8.8 Exemples.....</a>  | <a href="#">19</a> |

## 2 Syntaxe

```
ch2 [*] = CREA_CHAMP (

    ♦ TYPE_CHAM = /'NOEU_xxx' ,
                  /'CART_xxx' ,
                  /'ELNO_xxx' ,
                  /'ELGA_xxx' ,
                  /'ELEM_xxx' ,

    # pour imposer la numérotation des ddls si le champ résultat
    # est un cham_no :
    ◇ / NUME_DDL = nu , [nume_ddl]
      / CHAM_NO = cnonu , [cham_no]

    ◇ TITRE = titre , [l_Kn]
    ◇ INFO = /1 , [DEFAULT]
            /2 ,

    / OPERATION = 'AFFE' ,
    # =====

    ♦ / MAILLAGE = ma , [maillage]

    . / MODELE = mo , [modele]
      ◇ OPTION = option , [Kn]
      ◇ PROL_ZERO = / 'NON' , [DEFAULT]
                  / 'OUI' ,

    ♦ AFFE= (_F ( ♦ / TOUT = 'OUI' ,
                  / | GROUP_MA = grma , [l_grma]
                  | MAILLE = maille , [l_maille]
                  | GROUP_NO = grno , [l_grno]
                  | NOEUD = noeud , [l_noeud]

                  ♦ NOM_CMP = lcmp , [l_K8]
                  ♦ / VALE = lvaler , [l_R]
                  / VALE_I = lvalei , [l_I]
                  / VALE_C = lvalec , [l_C]
                  / VALE_F = lvalef , / [l_fonction]
                  / [l_formule]

                ),),
```

```

/  OPERATION =                'ASSE' ,
#  =====

  ◇ reuse =                ch2 ,                                [champ]

  ◆ /  MAILLAGE =                ma ,                                [maillage]
    /  MODELE =                mo ,                                [modele]
      ◇ OPTION =                option                                [Kn]
      ◇ PROL_ZERO =                /  'NON' ,                        [DEFAULT]
                                      /  'OUI' ,

  ◆ ASSE= (_F (  ◆ /  TOUT = 'OUI' ,
                  /  | GROUP_MA = grma , [l_grma]
                  | MAILLE = maille , [l_maille]
                  | GROUP_NO = grno , [l_grno]
                  | NOEUD = noeud , [l_noeud]

                  ◆ CHAM_GD = ch1 , [champ]

                  ◇ NOM_CMP = lcmp , [l_K8]
                    ◇ NOM_CMP_RESU = lcmp_resu, [l_K8]

                  ◇ /  CUMUL = 'NON' , [DEFAULT]
                    /  CUMUL = 'OUI' ,
                      ◇ /  COEF_R = /  1. , [DEFAULT]
                                      /  coefr , [R]
                                      /  COEF_C = coefc , [C]

                  ),),

/  OPERATION =                'COMB' ,
#  =====

  ◆ COMB= (_F (
      ◆ CHAM_GD = ch1 , [cham_no]
      ◆ COEF_R = coefr , [R]
  ),),

/  OPERATION =                'EVAL' ,
#  =====

  ◆ CHAM_F = ch_fonc , / [cham_no (NEUT_F)]
                                     / [cham_elem (NEUT_F)]
  ◆ CHAM_PARA= l_ch_para , [l_champ]

/  OPERATION =                'DISC' ,
#  =====
#  si le champ résultat est un CHAM_ELEM :
  ◇ MODELE = mo , [modele]
  ◇ OPTION = option [Kn]
  ◇ PROL_ZERO = /  'NON' , [DEFAULT]
                  /  'OUI' ,

  ◆ CHAM_GD = ch1 , [champ]

/  OPERATION =                'NORMALE' ,
#  =====

  ◆ MODELE = mo , [modele]
  ◆ |GROUP_MA = l_gma , [l_group_ma]
  |MAILLE = l_mail , [l_maille]

/  OPERATION =                'R2C' ,

```

```
# =====  
# pour transformer un champ réel en champ complexe  
# (avec partie imaginaire nulle)  
♦ CHAM_GD = chR , [champ]
```

```
/ OPERATION = 'C2R' ,
# =====
# pour transformer un champ complexe en champ réel :
♦ CHAM_GD = chC , [champ]
♦ PARTIE = / 'REEL', # partie réelle
           / 'IMAG', # partie imaginaire
           / 'MODULE', # valeur absolue
           / 'PHASE', # "phase" (en degrés)

/ OPERATION = 'EXTR' ,
# =====
# extraction du champ de géométrie d'un maillage :
♦ | MAILLAGE = ma , [maillage]
  ♦ NOM_CHAM = 'GEOMETRIE' ,

# extraction dans une table :
| TABLE = tabl , [table]
♦ / MAILLAGE = ma , [maillage]
  / MODELE = mo , [modele]
  ♦ OPTION = option , [kn]
  ♦ PROL_ZERO = / 'NON' , [DEFAULT]
                  / 'OUI' ,

# extraction des "level set" d'une SD fiss_xfem :
| FISSURE = fxfem , [fiss_xfem]
♦ NOM_CHAM = / 'LTNO' ,
              / 'LNNO' ,
              / 'GRLTNO' ,
              / 'GRLNNO' ,
              / 'STNO' ,
              / 'STNOR' ,
              / 'BASLOC' ,

# extraction d'un champ d'une SD_RESULTAT :
| RESULTAT = resu ,
♦ NOM_CHAM = / 'ACCE' ,
              / ... (cf. [§7.1]),

♦ SENSIBILITE = / theta, [theta_geom]
                 / para , [para_sensi]

♦ / # Sélection d'un numéro d'ordre dans la SD_RESULTAT
  / NUME_ORDRE = nuodr , [I]
  / NUME_MODE = numode , [I]
  / NOEUD_CMP = (noeud,cmp) , [l_K8]
  / NOM_CAS = nocas , [Kn]
  / ANGL = alpha , [R]
  / ♦ / INST = inst , [R]
      / FREQ = freq , [R]

      ♦ / CRITERE = / 'RELATIF', [DEFAULT]
          ♦ PRECISION = / prec , [R]
                        / 1.0E-6, [DEFAULT]
          / CRITERE = / 'ABSOLU',
          ♦ PRECISION = / prec , [R]

      ♦ INTERPOL = / 'NON' , [DEFAULT]
                  / 'LIN' ,
```

```

/ # Calcul d'un champ contenant les "extrema" d'une
# SD_RESULTAT
♦ TYPE_MAXI = / 'MAXI' ,
              / 'MINI' ,
              / 'MAXI_ABS' ,
              / 'MINI_ABS' ,
              / 'NORM_TRAN' ,

◇ TYPE_RESU = / 'VALE' , [DEFAULT]
              / 'INST' ,

◇ / TOUT_ORDRE = 'OUI' , [DEFAULT]
  / LIST_INST = linst , [listr8]
  / LIST_FREQ = lfreq , [listr8]

    ◇ / CRITERE = / 'RELATIF', [DEFAULT]
      ◇ PRECISION = / prec , [R]
        / 1.0E-6, [DEFAULT]
    / CRITERE = / 'ABSOLU',
      ♦ PRECISION = / prec , [R]

)

Si TYPE_CHAM = 'NOEU_xxx' alors [*] = cham_no
              'CART_xxx'      carte
              'ELNO_xxx'      cham_elem
              'ELGA_xxx'      cham_elem
              'ELEM_xxx'      cham_elem
```



## 3 Opérandes

### 3.1 Opérandes généraux

#### 3.1.1 Opérande TYPE\_CHAM, affectation d'un type au champ résultat

Ce mot clé (obligatoire) sert d'abord à typer le champ résultat de la commande. Il est formé de 2 "mots" reliés par un "blanc souligné" (\_) :

TYPE\_CHAM= 'XXXX\_GD' où :

|       |   |          |  |             |
|-------|---|----------|--|-------------|
| XXXX= | / | 'NOEU'   | Champ aux nœuds                        | (cham_no)   |
|       | / | 'CART'   | Champ constant par maille              | (carte)     |
|       | / | 'ELNO'   | Champ par éléments aux nœuds           | (cham_elem) |
|       | / | 'ELGA'   | Champ par éléments aux points de Gauss | (cham_elem) |
|       | / | 'ELEM'   | Champ constant par élément             | (cham_elem) |
| GD=   | / | 'DEPL_R' | déplacement                            |             |
|       | / | 'SIEF_R' | contrainte                             |             |
|       | / | 'TEMP_R' | température                            |             |
|       | / | 'FLUX_R' | flux                                   |             |
|       | / | ...      |  |             |

Le type du champ résultat est déduit de cette information donnée par l'utilisateur. Par exemple :

```
TYPE_CHAM= 'NOEU_DEPL_R'      ->   cham_no   (DEPL_R)
TYPE_CHAM= 'CART_SIEF_R'      ->   carte      (SIEF_R)
TYPE_CHAM= 'ELNO_EPSI_R'      ->   cham_elem  (EPSI_R)
TYPE_CHAM= 'ELGA_VARI_R'      ->   cham_elem  (VARI_R)
```

Ce mot clé sert aussi à préciser (pour la commande) quelle doit être la nature du champ voulu en résultat. Il est indispensable pour les opérations 'AFFE', 'ASSE' et 'DISC'.

Exemples :

```
OPERATION= 'AFFE' + TYPE_CHAM= 'CART_DEPL_R'  => une carte de DEPL_R.
OPERATION= 'ASSE' + TYPE_CHAM= 'NOEU_EPSI_R'  => un cham_no de EPSI_R.
OPERATION= 'DISC' + TYPE_CHAM= 'NOEU_SIEF_R'  => un cham_no de SIEF_R.
```

Il n'y a que deux opérations pour lesquelles ce mot clé est une contrainte inutile (mais obligatoire !) pour l'utilisateur (OPERATION='EVAL' et OPERATION='EXTR') car pour ces deux opérations, la nature du champ résultat est imposée par le choix de l'opération.

Le renseignement du mot clé TYPE\_CHAM est (malheureusement) fastidieux pour l'OPERATION='EXTR'. Il se déduit du mot clé NOM\_CHAM. La correspondance est donnée dans [§3.7.1].

#### Remarque importante

*La possibilité de créer des cham\_elem de n'importe quelle grandeur est conditionnée par l'état de développement (informatique) des types d'éléments finis du modèle. Tout n'est pas encore possible ; par exemple, pour créer un cham\_elem de FLUX\_R sur un modèle contenant les éléments DKT, il faut que cet élément fini ait prévu de le faire (ce qui n'est pas le cas aujourd'hui).*

*On ne peut donner ici la liste précise des grandeurs permises pour chaque type d'élément fini. On se contentera de dire approximativement :*

- *pour les éléments iso-paramétriques de mécanique, sont permises les grandeurs : GEOM\_R, INST\_R, NEUT\_R, NEUT\_F, EPSI\_R, SIEF\_R, VARI\_R, DOMMAG et HYDR\_R,*
- *pour les éléments iso-paramétriques de thermique, sont permises les grandeurs : GEOM\_R, INST\_R, NEUT\_R et NEUT\_F.*

## 3.1.2 Opérande MAILLAGE = mailla

Lorsque l'on crée un CHAM\_NO ou une CARTE, il faut préciser en général sur quel maillage s'appuiera ce champ. Pour cela, on utilise le mot clé MAILLAGE.

## 3.1.3 Opérands MODELE, OPTION, PROL\_ZERO

Lorsque l'on crée un CHAM\_ELEM, il faut préciser sur quels éléments finis sera défini le champ. Pour cela, on utilise le mot clé MODELE.

Pour décrire la structure d'un CHAM\_ELEM, il ne suffit pas de donner (via le MODELE) un type d'élément pour chaque maille, car un type d'élément peut connaître plusieurs "formes" pour une grandeur donnée. Pour créer le champ souhaité, l'utilisateur peut utiliser le mot clé OPTION. Si par exemple, il écrit : `X= CREA_CHAMP(... MODELE=mo, OPTION='SIEF_ELGA', ...)`, le champ créé par CREA\_CHAMP aura la même forme que s'il avait été calculé par CALC\_ELEM / OPTION='SIEF\_ELGA'. Si l'utilisateur n'emploie pas le mot clé OPTION, CREA\_CHAMP choisira (s'il le peut) une forme par défaut.

Remarque : pour l'opération 'ASSE', quand on assemble des cham\_elem, il vaut mieux en général ne pas fournir le mot clé OPTION. L'option qui sera choisie sera celle du champ fourni dans la 1ère occurrence du mot clé facteur ASSE si la grandeur associée à ce champ est la même que celle du champ résultat.

Dans un CHAM\_ELEM, tous les éléments d'un même type (même type de maille et même modélisation) doivent obligatoirement posséder les mêmes composantes. Si le champ résultat est incomplètement décrit (si l'on n'a pas affecté toutes les mailles), il faut autoriser (éventuellement) la commande CREA\_CHAMP à "prolonger" le champ par "zéro" là où il n'a pas été décrit. C'est l'objet du mot clé PROL\_ZERO.

Si PROL\_ZERO = 'NON' (défaut)

Si des éléments sont incomplets, on s'arrête en erreur fatale.

Si PROL\_ZERO = 'OUI'

On prolonge le champ incomplètement décrit par des "0"

## 3.1.4 Mots clé NUME\_DDL et CHAM\_NO

Ces deux mots clé permettent d'imposer une numérotation pour le champ résultat (si celui-ci est un CHAM\_NO). On donne via ces mots clé un "modèle" de numérotation pour le champ résultat.

Si on donne NUME\_DDL= nu, on prendra comme numérotation celle de nu. Cette possibilité n'est valable que pour les champs de déplacements (phénomène 'MECANIQUE') ou pour des champs de température (phénomène 'THERMIQUE') ou des champs de pression acoustique (phénomène 'ACOUSTIQUE').

Si on donne CHAM\_NO= chno, on prendra la numération de chno.

### Remarque sur l'espace disque utilisé :

Ces 2 mots clés permettent parfois d'économiser beaucoup de place sur la base "Globale". Quand par exemple, on extrait de nombreux cham\_no d'une SD\_RESULTAT, si on n'utilise pas l'un de ces mots clés, on duplique le profil du champ pour chacun d'eux. Si on utilise l'un de ces 2 mots clés, tous ces champs s'appuieront sur le profil contenu dans chno (ou nu).

## 3.1.5 Opérande TITRE = titr

Titre que l'on veut donner au champ résultat [U4.03.01].

## 3.1.6 Opérande INFO = /1 /2

INFO = 1  
Aucune impression.

INFO = 2  
Impression sur le fichier 'MESSAGE' du champ résultat.

## 3.2 Opérande OPERATION = /'AFFE' / 'ASSE' / 'EVAL' / 'DISC' / 'EXTR' / 'R2C' / 'C2R' / 'COMB'

Cet opérande sert à choisir le "mode" de fabrication du champ résultat. On peut créer un champ :

- par affectation de valeurs sur des nœuds ou des mailles (OPERATION='AFFE'),
- en assemblant des morceaux de champs définis sur des morceaux de maillages (OPERATION='ASSE'),
- en modifiant la représentation géométrique (discrétisation) d'un champ (passage nœuds <-> points de Gauss par exemple) (OPERATION='DISC'),
- en extrayant un champ d'une SD de type SD\_RESULTAT (evol\_ther, evol\_noli, mode\_meca, ...) (OPERATION='EXTR').
- en extrayant des valeurs numériques d'une table dont les colonnes ont des noms prédéfinis : 'MAILLE', 'NOEUD'...
- en combinant linéairement des champs (OPERATION='ASSE'),
- en "combinant" (multiplication, exponentielle, ...) des champs (OPERATION='EVAL'),
- en évaluant les fonctions d'un champ de fonctions pour en faire un champ de réels (OPERATION='EVAL'),
- en transformant un champ réel en champ complexe (ou réciproquement) (OPERATION='R2C' / 'C2R'),
- en faisant une combinaison linéaire de plusieurs `cham_no` ayant la même numérotation (OPERATION='COMB'). A la différence de l'opération 'ASSE', le champ résultat conservera les ddls de Lagrange associés à la dualisation des conditions aux limites.

## 3.3 Opérandes pour OPERATION = 'AFFE'

Cette opération permet d'affecter des valeurs (réelles, entières, complexes ou fonction) sur des entités géométriques (nœuds ou mailles) d'un maillage.

La grandeur associée au champ est implicitement donnée par le mot clé `TYPE_CHAM` (ci-dessus).

### 3.3.1 Mot clé facteur AFFE

Les opérandes sont regroupés sous le mot clé facteur `AFFE`. Ce mot clé est répétable. Le principe de surcharge est appliqué entre les différentes occurrences du mot clé `AFFE` : si une entité géométrique est affectée plusieurs fois, la dernière affectation l'emporte.

### 3.3.2 Opérandes TOUT='OUI', GROUP\_MA, GROUP\_NO, MAILLE et NOEUD

Les entités géométriques que l'on veut affecter sont données par les opérandes `TOUT='OUI'`, `GROUP_MA`, `GROUP_NO`, `MAILLE` et `NOEUD`.

Si `TYPE_CHAM='NOEU_XXX'`, on affecte des nœuds ; l'utilisation des opérandes `GROUP_MA` et `MAILLE` est possible et signifie que l'on affecte tous les nœuds des mailles spécifiées.

Si `TYPE_CHAM:'EL.._XXX'` (ou `'CART_XXX'`), on affecte des mailles ; l'utilisation des opérandes `GROUP_NO` et `NOEUD` est alors interdite.

### 3.3.3 Opérande NOM\_CMP

Les noms des composantes que l'on veut affecter sont donnés par l'opérande `NOM_CMP`.

### 3.3.4 Opérandes VALE, VALE\_I, VALE\_C ou VALE\_F

Les valeurs à affecter sont données par les opérandes VALE, VALE\_I, VALE\_C ou VALE\_F selon la nature (réel, entier, complexe, fonction (ou formule) ) des composantes de la grandeur (DEPL\_R : réel, DEPL\_C : complexe, TEMP\_F : fonction / formule , ....).

### 3.3.5 Remarque

La règle de rémanence (voir U1.03.00) s'applique pour les différentes composantes que l'on peut affecter.

### 3.3.6 Exemples

Création d'un champ aux nœuds de déplacement. On veut imposer la numérotation du champ (celle de cnomod) :

```
DEPL1 = CREA_CHAMP( OPERATION= 'AFFE',  
                    TYPE_CHAM='NOEU_DEPL_R' , MAILLAGE = MA , CHAM_NO= CNOMOD,  
                    AFFE= (  
                        _F (TOUT='OUI' , NOM_CMP=('DX','DY','DZ') , VALE=(0.,0.,0.)),  
                        _F (GROUP_MA=('GM1','GM2') , NOM_CMP= 'DX' , VALE= 3.5d-2),  
                        _F (NOEUD=('N5','N7','N9') , NOM_CMP= 'DY' , VALE= 1.6d-2),  
                    )  
)
```

Création d'une carte de température (fonctions) :

```
TEMPF = CREA_CHAMP( OPERATION= 'AFFE',  
                    TYPE_CHAM='CART_TEMP_F' , MAILLAGE = MA,  
                    AFFE= ( _F ( TOUT='OUI' , NOM_CMP=('TEMP') , VALE_F= F1),  
                        _F ( GROUP_MA=('GM1','GM2') , NOM_CMP= ('TEMP' ) , VALE_F= F2), )  
)
```

## 3.4 Opérandes pour OPERATION = 'ASSE'

### 3.4.1 Généralités

Cet opérateur "assemble" des "morceaux de champs" pour en fabriquer un nouveau. Chaque occurrence du mot clé ASSE définit un morceau de champ. On appelle un morceau de champ, la restriction d'un champ existant (carte / cham\_no ou cham\_elem) sur un ensemble d'entités géométriques (mailles ou nœuds) et sur un ensemble de composantes.

Il y a un principe de surcharge des occurrences du mot clé ASSE si les morceaux se recouvrent les uns les autres.

Actuellement, on peut fabriquer :

- un cham\_no en assemblant des morceaux de cham\_no.
- un cham\_elem en assemblant des morceaux de cham\_elem et/ou de cartes.

L'opération 'ASSE' permet également de changer la grandeur associée à un champ ; par exemple transformer un champ de déformation (EPSI\_R) en champ de contraintes (SIEF\_R). Pour cela il faut utiliser les mots clé NOM\_CMP et NOM\_CMP\_RESU.

L'assemblage des morceaux de champs peut se faire en cumulant les morceaux (mots clé CUMUL et COEF\_R). Cela permet d'utiliser cette commande pour faire des combinaisons linéaires de CHAM\_NO ou de CHAM\_ELEM.

Remarque : l'opération 'ASSE' est la seule pour laquelle le champ résultat peut être "réentrant". Cette possibilité a été introduite pour pouvoir faire " $U = U + dU$ ".

## 3.4.2 Opérandes MAILLAGE, MODELE et PROL\_ZERO

Même usage que pour OPERATION= 'AFFE' [§3.3.1] et [§3.3.2].

## 3.4.3 Opérandes pour le mot clé facteur ASSE

Chaque occurrence du mot clé facteur ASSE permet de définir un morceau de champ que l'on assemble dans le champ résultat.

### 3.4.3.1 Opérande CHAM\_GD = ch1

ch1 est le champ (existant) avec lequel on veut fabriquer un morceau de champ.

### 3.4.3.2 Opérandes TOUT='OUI', GROUP\_MA, GROUP\_NO, MAILLE et NOEUD

Ces opérandes servent à définir la restriction géométrique du champ ch1. Si ch1 est un CHAM\_NO, on peut utiliser tous ces opérandes. Si ch1 est un CHAM\_ELEM (ou une CARTE), on ne peut pas utiliser les opérandes GROUP\_NO et NOEUD.

### 3.4.3.3 Opérandes NOM\_CMP et NOM\_CMP\_RESU

L'opérande NOM\_CMP sert à définir les composantes sur lesquelles on veut restreindre le champ ch1. Si NOM\_CMP est absent, on prend toutes les composantes de ch1.

L'opérande NOM\_CMP\_RESU sert à renommer (si on le souhaite) les composantes de ch1. Si NOM\_CMP\_RESU est fourni, NOM\_CMP doit l'être aussi et les deux listes en correspondance doivent être de même longueur.

Exemple 1 : transformer un champ de EPSI\_R en champ de VARI\_R  
CHVARI=CREA\_CHAMP(OPERATION='ASSE', TYPE\_CHAM='ELGA\_VARI\_R',  
MODELE=MO,  
ASSE=\_F(CHAM\_GD=CHEPSI, TOUT='OUI',  
NOM\_CMP=('EPXX', 'EPYY'),  
NOM\_CMP\_RESU=('V3', 'V1'), ), )

Exemple 2 : permuter les cmps SIXX et SIYY d'un champ de SIEF\_R  
CHS2=CREA\_CHAMP(OPERATION='ASSE', TYPE='NOEU\_SIEF\_R',  
MAILLAGE= MA ,  
ASSE=\_F(CHAM\_GD=CHS1, TOUT='OUI',  
NOM\_CMP=('SIXX', 'SIYY'),  
NOM\_CMP\_RESU=('SIYY', 'SIXX'), ), )

### 3.4.3.4 Opérandes CUMUL, COEF\_R et COEF\_C

L'opérande CUMUL='OUI' veut dire que les valeurs de l'occurrence concernée seront additionnées aux éventuelles valeurs déjà existantes.

Si CUMUL='NON', la valeur affectée remplace la valeur éventuellement déjà présente (CUMUL='OUI' est invalide pour les champs de "texte" (k8/k16, ...) bien entendu).

L'opérande COEF\_R = coefr permet la multiplication du morceau de champ par le coefficient réel coefr avant de l'assembler au champ résultat.

Exemple :

Fabriquer le cham\_elem :  $ch3 = 2. * ch1 + 3. * ch2$

```
CH3= CREA_CHAMP( OPERATION= 'ASSE',  
MODELE = MO , TYPE_CHAM ='ELGA_EPSI_R',  
ASSE = (_F( CHAM_GD = CH1 , TOUT ='OUI',  
CUMUL='OUI', COEF_R = 2.),  
_F( CHAM_GD = CH2 , TOUT ='OUI',  
CUMUL='OUI', COEF_R = 3.), )  
)
```

**Remarque concernant les champs complexes**

Le mot clé `COEF_C` n'est accepté que dans le cas où le champ résultat ( `CH3` ) et les champs arguments ( `CH1` et `CH2` ) sont tous complexes. Pour faire une combinaison linéaire avec coefficients complexes de champs réels, il faut transformer au préalable les champs réels en champs complexes. Voir `OPERATION = 'R2C'`.

### 3.4.4 Exemples

**Exemple 1**

Fabriquer un `cham_no` de température en extrayant un champ déjà calculé (dans un `evol_ther`) et en le redéfinissant (à 25. degrés) sur le groupe de mailles `soudur1`.

```
CH1= CREA_CHAMP ( OPERATION= 'EXTR' , TYPE_CHAM='NOEU_TEMP_R',  
                  RESULTAT= EVOTH , NOM_CHAM= 'TEMP', INST = 12.)  
CH2= CREA_CHAMP ( OPERATION = 'AFFE', TYPE_CHAM='NOEU_TEMP_R',  
                  MAILLAGE =MA,  
                  AFFE=_F( TOUT = 'OUI' , NOM_CMP = 'TEMP', VALE = 25. ) )  
  
CH3= CREA_CHAMP ( OPERATION = 'ASSE',  
                  MAILLAGE = MA, TYPE_CHAM = 'NOEU_TEMP_R',  
                  ASSE = ( _F( CHAM_GD = CH1 , TOUT = 'OUI', ),  
                           _F( CHAM_GD = CH2 , GROUP_MA = SOUDUR1 ), )  
                  )
```

**Exemple 2 :**

Fabriquer un `cham_elem` de `VARI_R` (pour l'utiliser en tant qu'état initial pour `STAT_NON_LINE`) en récupérant les variables internes (6 et 8) d'une loi de comportement pour en faire les variables 1 et 2 de la (nouvelle) loi de comportement qui sera utilisée dans le `STAT_NON_LINE` à venir.

```
CH1= CREA_CHAMP ( OPERATION= 'EXTR', TYPE_CHAM='ELGA_VARI_R',  
                  RESULTAT= STNL, NOM_CHAM= 'VARI_ELGA', INST = 4.)  
  
CH2= CREA_CHAMP ( OPERATION= 'ASSE',  
                  MODELE = MO , TYPE_CHAM = 'ELGA_VARI_R',  
                  ASSE = _F( CHAM_GD = CH1 , TOUT = 'OUI',  
                             NOM_CMP = ( 'V6', 'V8' ),  
                             NOM_CMP_RESU = ( 'V1', 'V2' ), ) )
```

### 3.5 Opérandes pour le mot clé facteur COMB

Ce mot clé permet de calculer la combinaison linéaire de plusieurs `cham_no` ayant la même numérotation. A la différence de l'opération 'ASSE', le champ résultat contiendra également les coefficients de Lagrange correspondant à la dualisation des conditions aux limites.

Chaque occurrence du mot clé facteur ASSE permet de définir un élément de la combinaison linéaire.

#### 3.5.1 Opérande CHAM\_GD = ch1

`ch1` est le `cham_no` (existant) que l'on veut combiner linéairement.

#### 3.5.2 Opérande COEF\_R = coefr

`coefr` est le coefficient réel appliqué à `ch1` pour la combinaison.

#### 3.5.3 Exemple

Pour calculer  $C = 1.A - 2.B$ , on écrit :

```
C= CREA_CHAMP ( OPERATION='COMB', TYPE_CHAM='NOEU_DEPL_R',  
                COMB= (
```

```
      F(CHAM_GD=A, COEF_R= 1.),  
      F(CHAM_GD=B, COEF_R=-2.),  
    ))
```

## 3.6 Opérandes pour OPERATION = 'EVAL'

Cette opération sert à transformer un champ de fonctions en champs de réels en évaluant les fonctions du champ de fonctions.

Le champ de fonctions est obligatoirement un champ de la grandeur 'NEUT\_F' et le champ résultat sera toujours un champ de 'NEUT\_R'. Ce champ pourra être transformé en champ d'une autre grandeur quelconque en faisant appel une seconde fois à la commande CREA\_CHAMP / OPERATION='ASSE'.

Un exemple de l'usage de l'opération 'EVAL' est donné dans le document [U2.01.09] "Définition analytique d'un champ de contraintes ..."

### 3.6.1 Opérande CHAM\_F= chf

chf est le nom du champ de fonctions à évaluer (CHAM\_NO, CARTE ou CHAM\_ELEM).

### 3.6.2 Opérande CHAM\_PARA= l\_chpara

l\_chpara est la liste des champs "paramètres" pour l'évaluation des fonctions. Tous les champs de l\_chpara doivent être discrétisés de la même façon que chf. Par exemple, si chf est un CHAM\_ELEM/ELGA, il faut que tous les champs de l\_chpara soient aussi des CHAM\_ELEM/ELGA.

Il faut que la liste des champs paramètres soit suffisante pour permettre l'évaluation de toutes les fonctions référencées dans chf.

## 3.6.3 Exemples

### 3.6.3.1 Exemple 1

On veut créer un cham\_elem (SIEF\_R) aux points de Gauss dont les composantes soient des fonctions analytiques de la géométrie et du temps. Pour cet exemple, on suppose que l'on a déjà fabriqué deux champs aux points de Gauss CHGEOMG : champ de géométrie et CHINSTG : champ d'instant.

```
RHO=1000.  
G=10.  
KP=3.  
SIZZ = FORMULE (NOM_PARA = 'Z', VALE = 'RHO*G*Z')  
SIXX = FORMULE (NOM_PARA = ('Z','INST'), VALE = 'KP*SIZZ(Z)+3.*INST')  
  
# Affectation des fonctions :  
# -----  
SIG1=CREA_CHAMP(OPERATION='AFFE', TYPE_CHAM='ELGA_NEUT_F',  
  MODELE=MO , PROL_ZERO='OUI',,  
  AFFE=_F(TOUT='OUI', NOM_CMP=('X1','X2'),  
    VALE_F=('SIXX','SIZZ') ))  
  
# Evaluation des fonctions :  
# -----  
SIG2= CREA_CHAMP(OPERATION='EVAL' ,TYPE_CHAM='ELGA_NEUT_R',  
  MODELE=MO ,CHAM_F=SIG1 , CHAM_PARA=( CHGEOMG,CHINSTG) )  
  
# transformation du champ de NEUT_R en SIEF_R :
```

```
# -----
SIG3=CREA_CHAMP(OPERATION='ASSE', TYPE_CHAM='ELGA_SIEF_R',
  MODELE=MO , PROL_ZERO='OUI',
  ASSE=_F(TOUT='OUI', CHAM_GD=SIG2,
    NOM_CMP= ('X1' , 'X2' ),
    NOM_CMP_RESU= ('SIXX', 'SIZZ'),
  ))
```

### 3.6.3.2 Exemple 2

On veut calculer un champ de température aux nœuds ( *CH3* ) contenant le produit de 2 autres champs aux nœuds de température ( *CH1* et *CH2* )

```
# 1) transformation des cham_no/TEMP_R (CH1 et CH2) en cham_no/NEUT_R:
# -----
CH1N=CREA_CHAMP( OPERATION='ASSE', TYPE_CHAM='NOEU_NEUT_R', MAILLAGE=MA,
  ASSE=_F( TOUT = 'OUI', CHAM_GD = CH1,
    NOM_CMP = ('TEMP',), NOM_CMP_RESU = ('X1',)), )

CH2N=CREA_CHAMP( OPERATION='ASSE', TYPE_CHAM='NOEU_NEUT_R', MAILLAGE=MA,
  ASSE=_F( TOUT = 'OUI', CHAM_GD = CH2,
    NOM_CMP = ('TEMP',), NOM_CMP_RESU = ('X2',)), )

# 2) multiplication    CH3N = CH1N * CH2N :
# -----
FMULT = FORMULE(NOM_PARA = ('X1', 'X2'), VALE = 'X1*X2')

CHFMU=CREA_CHAMP( OPERATION='AFFE', TYPE_CHAM='NOEU_NEUT_F', MAILLAGE=MA,
  AFFE=_F( TOUT = 'OUI', NOM_CMP = 'X3', VALE_F = FMULT) )

CH3N=CREA_CHAMP( OPERATION='EVAL', TYPE_CHAM='NOEU_NEUT_R',
  CHAM_F=CHFMU, CHAM_PARA=(CH1N,CH2N,) )

# 3) transformation du cham_no/NEUT_R (CH3N) en cham_no/TEMP_R:
# -----
CH3=CREA_CHAMP( OPERATION='ASSE', TYPE_CHAM='NOEU_TEMP_R', MAILLAGE=MA,
  ASSE=_F( TOUT = 'OUI', CHAM_GD = CH3N,
    NOM_CMP = ('X3',), NOM_CMP_RESU = ('TEMP',)), )
```

## 3.7 Opérandes pour OPERATION = 'DISC'

Cette opération sert à modifier la "discrétisation" d'un champ existant. Par exemple, transformer un champ aux nœuds en champs aux points de Gauss.

Les deux champs (donnée et résultat) sont associés à la même grandeur. Il existe une exception : le cas de l'opération **DISC** qui permet de transformer un `cham_no_VAR2_R` (ou une `carte_VAR2_R`) en `cham_elem_VARI_R` (voir par exemple le test `hplp100b`).

La discrétisation voulue par l'utilisateur pour son champ résultat est indiquée par le mot clé `TYPE_CHAM`.

Seuls les cas de figure suivants sont traités actuellement par la commande :



|                  |                     |
|------------------|---------------------|
| CARTE            | -> CHAM_ELEM / ELNO |
| CARTE            | -> CHAM_ELEM / ELGA |
| CARTE            | -> CHAM_ELEM / ELEM |
| CARTE            | -> CHAM_NO          |
| CHAM_NO          | -> CHAM_ELEM / ELNO |
| CHAM_NO          | -> CHAM_ELEM / ELGA |
| CHAM_NO          | -> CHAM_ELEM / ELEM |
| CHAM_ELEM / ELNO | -> CHAM_NO          |
| CHAM_ELEM / ELGA | -> CHAM_ELEM / ELNO |
| CHAM_ELEM / ELGA | -> CHAM_NO          |

Les ingrédients des traitements sont :

CARTE -> ELxx :

- La valeur (unique) portée par une maille est recopiée sur tous les points de la maille.

NOEU -> ELxx :

- le passage des valeurs des nœuds aux points internes de la maille se fait en utilisant les fonctions de forme des éléments finis du modèle.

ELGA -> ELNO :

- le passage des valeurs des points internes aux nœuds de la maille se fait en utilisant la matrice d'extrapolation Gauss->Nœuds.

ELNO -> NOEU :

- le passage aux valeurs des nœuds du maillage se fait par moyenne arithmétique des valeurs portées par les nœuds des éléments concourants.

### 3.7.1 Opérande CHAM\_GD= ch1

ch1 est le champ dont on veut modifier la "discrétisation".

### 3.7.2 mots clé MODELE et PROL\_ZERO

Même usage que pour OPERATION = 'AFFE' [3.3.2]

### 3.7.3 Exemple

```
# CHXG = CHAMP DE GEOMETRIE AUX POINTS DE GAUSS :  
# -----  
CHXN =CREA_CHAMP(OPERATION='EXTR', TYPE_CHAM='NOEU_GEOM_R',  
                 NOM_CHAM='GEOMETRIE', MAILLAGE=MA )  
CHXG= CREA_CHAMP(OPERATION='DISC', TYPE_CHAM='ELGA_GEOM_R',  
                 MODELE=MO , CHAM_GD= CHXN )
```

## 3.8 Opérandes pour OPERATION = 'NORMALE'

Cette opération sert à calculer les vecteurs "normaux" aux facettes d'un modèle. L'utilisateur doit indiquer avec les mots clés MODELE, GROUP\_MA et MAILLE le nom du modèle concerné ainsi que les "facettes" dont il souhaite calculer les normales. Les "facettes" peuvent être des éléments de "peau" d'un maillage 3D ou des éléments de plaque / coque.

Pour les maillages 2D, les "facettes" sont des éléments linéiques.

Le champ produit est un cham\_no (grandeur GEOM\_R) dont les composantes sont nommées X, Y, Z. La normale portée par un nœud est obtenue en moyennant les normales des facettes concourant en ce nœud. Le vecteur "normal" est de longueur 1.

## 3.9 Opérandes pour OPERATION = 'EXTR'

Cette opération sert en général à extraire un champ d'une SD de type resultat\_sdaster.

Il y a 3 exceptions :

- 1) on peut extraire le champ de géométrie des nœuds d'un maillage. Il faut alors utiliser les mots clés : MAILLAGE = ma, NOM\_CHAM = 'GEOMETRIE' et TYPE\_CHAM = 'NOEU\_GEOM\_R'.
- 2) on peut créer un champ en extrayant d'une table les valeurs correspondant à des paramètres de noms pré-établis : MAILLE, NOEUD, POINT, SOUS\_POINT, noms des composantes.
- 3) on peut extraire les "level set" associées à une fissure XFEM. Il faut alors utiliser les mots clés :

```
FISSURE = fiss_xfem,    NOM_CHAM =    / 'LTNO'      / 'LNNO'
                                   / 'GRLTNO'   / 'GRLNNO'
                                   / 'STNO'     / 'STNOR'
                                   / 'BASLOC'
```

et TYPE\_CHAM = 'NOEU\_NEUT\_R' excepté 'STNO' pour lequel ce sera 'NOEU\_NEUT\_I'

|          |  |
|----------|--|
| 'LTNO'   | level set tangente   |
| 'LNNO'   | level set normale  |
| 'GRLTNO' | gradient de la level set tangente  |
| 'GRLNNO' | gradient de la level set normale   |
| 'STNO'   | statut des nœuds, 1 valeur entière   |
| 'STNOR'  | statut des nœuds, 1 valeur réelle  |
| 'BASLOC' | base locale en fond de fissure, avec 6/9 valeurs réelles, 2/3 coordonnées pour le point et 2/3 coordonnées pour chacun des vecteurs de la base locale en 2D/3D |

### 3.9.1 Typage du champ résultat, mot clé TYPE\_CHAM

Le mot clé TYPE\_CHAM (obligatoire) [§3.2] doit être renseigné. Mis à part le cas de l'extraction dans une SD de type DYNA\_HARMO, le mot clé TYPE\_CHAM se déduit du nom symbolique du champ extrait (NOM\_CHAM). Le tableau ci-dessous donne la correspondance entre ces deux mots clés.

| NOM_CHAM         | TYPE_CHAM     | NOM_CHAM         | TYPE_CHAM     |
|------------------|---------------|------------------|---------------|
| 'ACCE'           | 'NOEU_DEPL_R' | 'INTE_ELNO'      | 'ELNO_INTE_R' |
| 'ACCE_ABSOLU'    | 'NOEU_DEPL_R' |                  |               |
| 'DERA_ELGA'      | 'ELGA_DERA_R' | 'INTE_NOEU'      | 'NOEU_INTE_R' |
| 'DERA_ELNO'      | 'ELNO_DERA_R' |                  |               |
| 'DEGE_ELNO'      | 'ELNO_EPSI_R' | 'IRRA'           | 'NOEU_IRRA_R' |
| 'DEPL'           | 'NOEU_DEPL_R' | 'META_ELGA_TEMP' | 'ELGA_VARI_R' |
| 'DEPL_ABSOLU'    | 'NOEU_DEPL_R' | 'META_ELNO'      | 'ELNO_VARI_R' |
| 'ECIN_ELEM'      | 'ELEM_ENER_R' | 'META_NOEU'      | 'NOEU_VAR2_R' |
| 'EFCA_ELNO'      | 'ELNO_SIEF_R' | 'PRAC_ELNO_ '    | 'ELNO_PRAC_R' |
| 'EFGE_ELNO'      | 'ELNO_SIEF_R' |                  |               |
| 'EFCA_NOEU'      | 'NOEU_SIEF_R' |                  |               |
| 'EFGE_NOEU'      | 'NOEU_SIEF_R' | 'PRAC_NOEU'      | 'NOEU_PRAC_R' |
| 'ENDO_ELNO_SIGA' | 'ELNO_SIEF_R' | 'PRME_NOEU'      | 'NOEU_PRME_R' |
| 'ENDO_ELNO_SIGM' | 'ELNO_SIEF_R' |                  |               |
| 'ENDO_ELNO_SINO' | 'ELNO_SIEF_R' |                  |               |
| 'EPME_ELGA'      | 'ELGA_EPSI_R' |                  |               |
| 'EPME_ELNO'      | 'ELNO_EPSI_R' | 'REAC_NODA'      | 'NOEU_DEPL_R' |
| 'EPME_ELNO_DPGE' | 'ELNO_EPSI_R' | 'SIEF_ELGA'      | 'ELGA_SIEF_R' |
| 'EPMG_ELGA'      | 'ELGA_EPSI_R' | 'SIEF_ELGA'      | 'ELGA_SIEF_R' |
| 'EPMG_ELNO'      | 'ELNO_EPSI_R' | 'SIEF_ELNO'      | 'ELNO_SIEF_R' |
| 'EPOT_ELEM'      | 'ELEM_ENER_R' | 'SIEF_ELNO'      | 'ELNO_SIEF_R' |

|                   |                 |                  |               |
|-------------------|-----------------|------------------|---------------|
| 'EPSA_ELNO'       | 'ELNO_EPSI_R'   | 'SICA_ELNO'      | 'ELNO_SIEF_R' |
| 'EPSG_ELGA'       | 'ELGA_EPSI_R'   | 'SIGM_ELNO'      | 'ELNO_SIEF_R' |
| 'EPSG_ELNO'       | 'ELNO_EPSI_R'   | 'SIGM_ELNO_SIEF' | 'ELNO_SIEF_R' |
| 'EPSI_ELGA'       | 'ELGA_EPSI_R'   | 'SIGM_ELNO_VARI' | 'ELNO_SIEF_R' |
| 'EPSI_ELNO'       | 'ELNO_EPSI_R'   | 'SICA_NOEU'      | 'NOEU_SIEF_R' |
| 'EPSI_NOEU'       | 'NOEU_SIEF_R'   | 'SIGM_NOEU'      | 'NOEU_SIEF_R' |
| 'EPSP_ELGA'       | 'ELGA_EPSI_R'   | 'SIGM_NOEU_SIEF' | 'NOEU_SIEF_R' |
| 'EPSP_ELNO'       | 'ELNO_EPSI_R'   | 'SIGM_NOEU_VARI' | 'NOEU_SIEF_R' |
| 'EPEQ_ELGA'       | 'ELGA_SIEF_R'   | 'SIZ1_NOEU'      | 'NOEU_SIEF_R' |
| 'SIEQ_ELGA'       | 'ELGA_SIEF_R'   | 'SIZ2_NOEU'      | 'NOEU_SIEF_R' |
| 'EPEQ_ELNO'       | 'ELNO_SIEF_R'   |                  | 'ELNO_SIEF_R' |
| 'SIEQ_ELNO'       | 'ELNO_SIEF_R'   |                  | 'ELNO_SIEF_R' |
| 'EPEQ_NOEU'       | 'NOEU_SIEF_R'   |                  | 'NOEU_SIEF_R' |
| 'SIEQ_NOEU'       | 'NOEU_SIEF_R'   | 'SOUR_ELGA'      | 'ELGA_SOUR_R' |
| 'ERRE_ELEM_NOZ1'  | 'ELEM_ERREUR'   | 'TEMP'           | 'NOEU_TEMP_R' |
| 'ERRE_ELEM_NOZ2'  | 'ELEM_ERREUR'   | 'THETA'          | 'NOEU_DEPL_R' |
| 'ERRE_ELGA_NORE'  | 'ELGA_ERREUR'   | 'SPMX_ELGA'      | 'ELEM_SPMA_R' |
| 'ERRE_ELNO_ELGA'  | 'ELNO_ERREUR'   | 'VARI_ELGA'      | 'ELGA_VARI_R' |
| 'FERRAILLAGE'     | 'ELEM_FER2_R'   |                  |               |
| 'FLUX_ELGA'       | 'ELGA_FLUX_R'   | 'VARI_ELNO'      | 'ELNO_VARI_R' |
| 'FLUX_ELNO'       | 'ELNO_FLUX_R'   | 'VARI_ELNO'      | 'ELNO_VARI_R' |
| 'FLUX_NOEU'       | 'NOEU_FLUX_R'   | 'VARI_NOEU'      | 'NOEU_VAR2_R' |
| 'FORC_NODA'       | 'NOEU_DEPL_R'   | 'VITE'           | 'NOEU_DEPL_R' |
| 'GEOMETRIE'       | 'NOEU_GEOM_R'   | 'VITE_ABSOLU'    | 'NOEU_DEPL_R' |
| 'GRAD_ELGA_THETA' | 'ELGA_G_DEPL_R' | 'VNOR_ELEM_DEPL' | 'ELEM_VNOR_C' |
| 'GRAD_ELNO_ELGA'  | 'ELNO_G_DEPL_R' |                  |               |
| 'GRAD_NOEU_THETA' | 'NOEU_VAR2_R'   |                  |               |
| 'HYDR_ELGA'       | 'ELGA_HYDR_R'   |                  |               |

Pour les DYNA\_HARMO, NOM\_CHAM peut prendre trois valeurs : 'DEPL', 'VITE' et 'ACCE'. Dans les trois cas, le type du champ résultat est un CHAM\_NO/DEPL\_C et il faut donc renseigner : TYPE\_CHAM='NOEU\_DEPL\_C'.

### 3.9.2 Opérande TABLE

◆ TABLE = tabl

Nom de concept `table` contenant les valeurs à stocker dans le champ. Les noms des paramètres de la table doivent respecter certaines règles.

Si le champ à produire est un champ aux nœuds 'NOEU\_XXXX', on doit trouver comme noms des paramètres de la table :

NOEUD + des noms de CMPS parmi les noms des CMPS de la grandeur XXXX.

Le paramètre NOEUD contient le nom du nœud.

Si le champ à produire est un champ par éléments 'ELGA\_XXXX' (ou 'ELNO\_XXXX'), on doit trouver comme noms des paramètres de la table :

MAILLE, POINT, [SOUS\_POINT] + des noms de CMPS parmi les noms des CMPS de la grandeur XXXX.

Le paramètre MAILLE contient le nom de la maille.

Le paramètre POINT contient le numéro du point de Gauss (ou du noeud) dans la maille.

Le paramètre SOUS\_POINT (nécessaire uniquement pour les champs à "sous-points") contient le numéro du sous-point dans le point de Gauss (ou le noeud).

Si le champ à produire est un champ constant par éléments 'ELEM\_XXXX', on doit trouver comme noms des paramètres de la table :

MAILLE, [SOUS\_POINT] + des noms de CMPS parmi les noms des CMPS de la grandeur XXXX.

### 3.9.3 Opérande RESULTAT

◆ RESULTAT = resu

Nom de concept `resultat` dans lequel on veut récupérer un champ.

### 3.9.4 Opérande `NOM_CHAM`

- ♦ `NOM_CHAM`

Ce mot clé précise le nom symbolique du champ à extraire [U4.71.00].

### 3.9.5 Opérande `SENSIBILITE = para`

Ce mot clé signifie que l'on ne souhaite pas extraire le véritable champ de la `SD_RESULTAT`, mais sa dérivée par rapport au paramètre sensible `para`.

Le document [U4.50.02] précise le fonctionnement de ce mot clé.

### 3.9.6 Opérands `NUME_ORDRE / NUME_MODE / NOM_MODE / ... / INTERPOL`

Ces mots clés servent à préciser quel est le numéro d'ordre de la `SD_RESULTAT` que l'on veut extraire.

Le choix des mots clés à utiliser dépend du type de `SD_RESULTAT` [U4.71.00].

#### Remarques

Lorsque l'on utilise `INTERPOL = 'LIN'`, le champ extrait sera une interpolation entre deux champs de la `SD_RESULTAT`. Cette interpolation n'a pas toujours de sens "physique"; par exemple sur des modes propres. Ce mot-clé ne devrait être utilisé que pour les `SD_RESULTAT` de type "`evol_xxxx`".

Lorsque l'on utilise un accès de type "réel" (`INST` ou `FREQ`), on cherche un champ dans un intervalle donné. Si on trouve plusieurs champs dans l'intervalle, le programme s'arrête en erreur fatale.

### 3.9.7 Calcul d'un champ contenant les "extrema" d'une `SD_RESULTAT`

L'idée est de créer un champ contenant en chaque point de l'espace la valeur extrême rencontrée au cours d'un transitoire (ou l'instant auquel cette valeur a été rencontrée).

Aujourd'hui ce paragraphe ne concerne que les résultats de type `evol_ther`, `evol_elas`, `evol_noli` et `dyna_trans`. Les champs sont toujours de type "réel".

On doit préciser :

- les numéros d'ordre définissant le transitoire : `TOUT_ORDRE`, `LIST_INST`,
- le type d'extrémalité choisi : `TYPE_MAXI` = '`MAXI`', ..., '`MINI_ABS`', '`NORM_TRAN`',
- ce que l'on veut : la valeur extrême ou l'instant où cette valeur est atteinte (`TYPE_RESU`).

#### 3.9.7.1 Opérande `TYPE_MAXI`

- / '`MAXI`' on considère le maximum atteint par les composantes au cours du transitoire,
  - / '`MAXI_ABS`' on considère le maximum atteint par la valeur absolue des composantes au cours du transitoire,
  - / '`MINI`' / '`MINI_ABS`' idem pour les minimums,
  - / '`NORM_TRAN`' on considère le maximum atteint par la quantité :  
 $DX^{*2} + DY^{*2} + DZ^{*2}$ .
- pour les 4 valeurs : '`MAXI`', ..., '`MINI_ABS`', les composantes du champ sont traitées indépendamment les unes des autres : les valeurs extrêmes peuvent ne pas être atteintes au même instant,
  - la cinquième valeur possible : '`NORM_TRAN`' n'est possible que pour les champs de `depl_R`. En un point donné, on cherche l'instant où la norme du vecteur translation est maximum et on recopie dans le champ résultat toutes les composantes du champ à l'instant trouvé.

#### 3.9.7.2 Opérande `TYPE_RESU`

/ 'VALE' , le champ résultat contient les valeurs extrêmes rencontrées au cours du transitoire,

**Remarque :**

Même si l'extremum a été obtenu avec une valeur absolue ( 'MAXI\_ABS' ou 'MINI\_ABS' ), la valeur stockée est algébrique.

/ 'INST' , le champ résultat contient les valeurs des instants où les valeurs extrêmes ont été rencontrées.

**Attention :** si par exemple :

```
NOM_CHAM = 'FLUX_ELNO',  
TYPE_RESU = 'INST',
```

Le champ résultat est un `cham_no/FLUX_R` qui contient des valeurs d'instants !

### 3.9.7.3 Opérandes TOUT\_ORDRE / LIST\_INST / LIST\_FREQ / PRECISION / CRITERE

Ces mots clés permettent de préciser l'étendue du transitoire à examiner.

Si `TOUT_ORDRE = 'OUI'` on passe en revue tous les numéros d'ordre.

Si `LIST_INST = linst` on ne considère que les instants spécifiés.

## 3.9.8 Exemples

### 3.9.8.1 Extraction d'un champ de température d'un concept resultat de type evol\_ther

```
temp10 = CREA_CHAMP ( OPERATION='EXTR',  
NOM_CHAM = 'TEMP' , TYPE_CHAM = 'NOEU_TEMP_R',  
RESULTAT = evoth , INST = 10. )
```

`temp10` est le champ de température extrait du résultat `evoth` (de type `evol_ther`) à l'instant 10.

### 3.9.8.2 Extraction d'un champ de déplacement d'un concept resultat de type mode\_meca

```
mode4 = CREA_CHAMP ( OPERATION='EXTR',  
NOM_CHAM = 'DEPL' , TYPE_CHAM = 'NOEU_DEPL_R',  
RESULTAT = modes , NUME_MODE = 4 )
```

`mode4` est le 4<sup>ème</sup> mode propre du résultat `modes` (de type `mode_meca`).

### 3.9.8.3 Extraction du champ de "température" contenant les instants où la température maximum a été atteinte au cours d'un transitoire

```
instmax = CREA_CHAMP ( OPERATION='EXTR',  
NOM_CHAM = 'TEMP' , TYPE_CHAM = 'NOEU_TEMP_R',  
RESULTAT = evoth ,  
TYPE_MAXI = 'MAXI' , TYPE_RESU = 'INST' )
```

### 3.9.8.4 Extraction d'un champ de contraintes dans une table

Soit le fichier (fort.81) contenant l'image de la table suivante :

```
#DEBUT_TABLE  
#TITRE CONTRAINTE `ELNO'  
MAILLE POINT SIXX SIYY SIZZ  
K8 I R R R  
M1 1 -1.632E+03 -2.553E+02 6.788E-01
```

```
      M1      2      -5.302E+03  -9.663E+01  6.018E+01
      M1      3      -3.638E+03  -1.058E+02  5.669E+01
      M2      1       5.632E+01   1.553E+02  3.788E-01
#FIN_TABLE
```

On peut extraire un champ de "contraintes" aux nœuds de cette table en faisant:

```
# lecture de la table :
TA=LIRE_TABLE(UNITE=81, TYPE_TABLE='TABLE', SEPARATEUR=' ')

# extraction du champ dans la table :
CH=CREA_CHAMP(OPERATION='EXTR', TYPE_CHAM='ELNO_SIEF_R', TABLE='TA',
              MODELE=MO, PROL_ZERO='OUI', OPTION='SIEF_ELNO_DEPL')
```

### 3.9.8.5 Calcul du champ des "normales" sur un groupe de mailles de bord

```
nor_DNOR = CREA_CHAMP( TYPE_CHAM = 'NOEU_GEOM_R',
                       OPERATION= 'NORMALE', MODELE= MO, GROUP_MA= 'FE' );
```