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 ⇔ points de Gauss par exemple),
- en extrayant un champ d'une structure de données de type SD_RESULTAT (evol_ther, evol_noli, mode_meca, ...).
- à partir des valeurs contenues dans 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

Remarque : La commande n'est pas ré-entrante sauf pour les opérations ASSE et COMB.
Dans ce cas, le mot-clé CHAM_GD est obligatoire.

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

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 CHAM_GD....9
3.1.3 Opérande MAILLAGE = mailla....9
3.1.4 Opérande PROL_ZERO....9
3.1.5 Opérandes MODELE, OPTION....9
3.1.6 Mots clés AFFE_SP / CARA_ELEM....9
3.1.7 Mots clé NUME_DDL et CHAM_NO....9
3.1.8 Opérande TITRE = titr....10
3.1.9 Opérande INFO = /1 /2....10
3.2 Opérande OPERATION = /'AFFE' / 'ASSE' / 'EVAL' / 'DISC' / 'EXTR' / 'R2C' / 'C2R' / 'COMB'. .10
3.3 Opérandes pour OPERATION = 'AFFE'....10
3.3.1 Mot clé facteur AFFE....11
3.3.2 Opérandes TOUT='OUI', GROUP_MA et GROUP_NO....11
3.3.3 Opérande NOM_CMP....11
3.3.4 Opérandes VALE, VALE_I, VALE_C ou VALE_F....11
3.3.5 Remarque....11
3.3.6 Exemples....11
3.4 Opérandes pour OPERATION = 'ASSE'....12
3.4.1 Généralités....12
3.4.2 Opérandes MAILLAGE, MODELE....12
3.4.3 Opérandes pour le mot clé facteur ASSE....12
3.4.4 Exemples....13
3.5 Opérandes pour le mot clé facteur COMB....14
3.5.1 Opérande CHAM_GD = ch1....14
3.5.2 Opérande COEF_R = coefr....14
3.5.3 Exemple....14
3.6 Opérandes pour OPERATION = 'EVAL'....14
3.6.1 Opérande CHAM_F= chf....14
3.6.2 Opérande CHAM_PARA= l_chpara....14
3.6.3 Exemples....15
3.7 Opérandes pour OPERATION = 'DISC'....16
3.7.1 Opérande CHAM_GD= ch1....17
3.7.2 mot clé MODELE....17
3.7.3 Exemple....17
3.8 Opérandes pour OPERATION = 'NORMALE'....17

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

3.9 Opérandes pour OPERATION = 'EXTR'....17
3.9.1 Typage du champ résultat, mot clé TYPE_CHAM....19
3.9.2 Opérande TABLE....21
3.9.3 Opérande RESULTAT....21
3.9.4 Opérande NOM_CHAM....21
3.9.5 Opérandes NUME_ORDRE / NUME_MODE / NOM_MODE / ... / INTERPOL....21
3.9.6 Calcul d'un champ contenant les "extrema" d'une SD_RESULTAT....22
3.9.7 Exemples....23

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

2 Syntaxe

```
ch2 [*] = CREA_CHAMP   (

      ◊  CHAM_GD   = ch1  ,                                    [champ]

      ◊  TITRE =   titre  ,                                    [l_Kn]
      ◊  INFO  =   /1 ,                                        [DEFAUT]
                   /2 ,
      ♦  TYPE_CHAM =  /'NOEU_xxx' ,
                      /'CART_xxx' ,
                      /'ELNO_xxx' ,
                      /'ELGA_xxx' ,
                      /'ELEM_xxx' ,

      # Pour imposer la numérotation du champ résultat
      # (uniquement permis pour les cham_no) :
      ◊   /   NUME_DDL = nu,                                   [nume_ddl]
          /  CHAM_NO = cnonu ,                                 [cham_no]

      # Si la "structure" du champ résultat(cham_elem ou cham_no)
      # exige des valeurs que l'on ne peut pas déterminer :
      ◊   PROL_ZERO =      /   'NON' ,                         [DEFAUT]
                           /   'OUI' ,

      # Si l'on souhaite créer un cham_elem à "sous-points" pour lequel
      # tous les sous-points (d'un même point) ont la même valeur :
      ◊   AFFE_SP= _F   (  ♦   CARA_ELEM    =   carele ,),     [cara_elem]


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

         ♦   /   MAILLAGE  =         ma ,                      [maillage]

      .      /   MODELE    =         mo ,                      [modele]

         ♦   AFFE= (_F (   ♦   /   TOUT      =   'OUI' ,
                               /      |   GROUP_MA  =   grma   ,   [l_grma]
                                      |   GROUP_NO  =   grno   ,   [l_grno]

                           ♦   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]
                       ),),
```


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

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

    ♦  /  MAILLAGE  =         ma ,                    [maillage]
       /  MODELE    =         mo ,                    [modele]
          ◊  OPTION    =      option                  [Kn]

    ♦  ASSE= (_F (  ♦      /  TOUT      =  'OUI' ,
                            /  |  GROUP_MA  =  grma   ,  [l_grma]
                               |  GROUP_NO  =  grno   ,  [l_grno]

                       ♦  CHAM_GD      = ch1  ,         [champ]

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

                       ◊  /  CUMUL  =         'NON'  ,  [DEFAUT]
                          /  CUMUL  =         'OUI'  ,
                             ◊  /  COEF_R =   /  1. ,   [DEFAUT]
                                              /  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]

    ♦  CHAM_GD   =      ch1,                        [champ]

/   OPERATION =            'NORMALE' ,
#   ===================================
    ♦  MODELE          =         mo ,                [modele]
    ♦  GROUP_MA        =         l_gma ,             [l_group_ma]
```


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

```
/   OPERATION =            'R2C' ,
#   ================================
    # pour transformer un champ aux nœuds réel en champ complexe
    # (avec partie imaginaire nulle)
    ♦  CHAM_GD             =         chR ,              [cham_no]

/   OPERATION =            'C2R' ,
#   ================================
    # pour transformer un champ aux nœuds complexe en champ réel :
    ♦  CHAM_GD      =         chC ,                     [cham_no]
    ♦  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 du champ d'abscisse curviligne d'un maillage :
       /  ♦  MAILLAGE =   ma ,                          [maillage]
          ♦  NOM_CHAM =   'ABSC_CURV' ,

       #  extraction dans une table :
       /  ♦   TABLE    =  tabl   ,                      [table]
          ♦  /  MAILLAGE  =  ma  ,                      [maillage]
             /  MODELE =  mo  ,                         [modele]
                ◊  OPTION =  option  ,                  [kn]

       #  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 cara_elem :
       /  ♦  CARA_ELEM =  carele ,                      [cara_elem]
          ♦  NOM_CHAM =   nomch ,                       [Kn]

       #  extraction d'un champ d'une SD char_meca :
       /  ♦  CHARGE    =  charge ,                      [char_meca]
          ♦  NOM_CHAM  =  nomch ,                       [Kn]
```


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

```
#  extraction d'un champ d'une SD_RESULTAT :
             /  ♦  RESULTAT =   resu ,
                ♦  NOM_CHAM =   /  'ACCE' ,
                                /  ... (cf. [3.9.4]),

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

                    # Si INST ou FREQ présents
                         ◊  /  CRITERE   =  / 'RELATIF', [DEFAUT]
                               ◊  PRECISION =  / prec ,  [R]
                                               / 1.0E-6, [DEFAUT]
                            /  CRITERE   =  / 'ABSOLU',
                               ♦  PRECISION =   prec ,  [R]

                   # Si INST présent
                         ◊  INTERPOL     =  / 'NON'   ,  [DEFAUT]
                                            / 'LIN'   ,

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

                       # si TYPE_MAXI vaut MAXI_ABS ou MINI_ABS
                      ◊  TYPE_RESU =  /  'VALE' ,            [DEFAUT]
                                      /  'INST' ,
                                      /  'VALE_ABS',
                       # sinon
                      ◊  TYPE_RESU =  /  'VALE' ,            [DEFAUT]
                                      /  'INST' ,

                      ◊  /  TOUT_ORDRE = 'OUI'  ,            [DEFAUT]
                         /  LIST_INST  = linst  ,            [listr8]
                         /  LIST_FREQ  = lfreq  ,            [listr8]

                            ◊  /  CRITERE   =  / 'RELATIF', [DEFAUT]
                                  ◊  PRECISION =  / prec ,  [R]
                                                  / 1.0E-6, [DEFAUT]
                               /  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
```


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

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 [3.9.1].

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.*

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

3.1.2 Opérande CHAM_GD

Avec les opérations ASSE et COMB, on peut réutiliser un champ existant. Dans ce cas, il faut indiquer ici quel champ est réutilisé.

3.1.3 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.

3.1.4 Opérande PROL_ZERO

Lorsqu'on crée un CHAM_ELEM, les valeurs existant dans le champ sont déterminées par les éléments finis du modèle. Par exemple, un champ SIGM_ELGA sur un modèle 2D **doit** contenir les 4 composantes SIXX, SIYY, SIZZ et SIXY.
Si la construction du champ ne permet pas de calculer toutes les valeurs attendues, on est confronté à un problème. Si le mot clé PROL_ZERO vaut 'OUI', les valeurs manquantes seront mises à zéro.
Si le mot clé PROL_ZERO vaut 'NON', le code s'arrêtera en erreur.

Ce problème concerne également (mais plus rarement) les CHAM_NO quand on veut imposer la numérotation de leurs composantes (voir les mots clés NUME_DDL et CHAM_NO).

3.1.5 Opérandes MODELE, OPTION

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_CHAMP / CONTRAINTE='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.

3.1.6 Mots clés AFFE_SP / CARA_ELEM

Le mot clé facteur (non répétable) AFFE_SP permet de demander la création d'un cham_elem à « sous-points » (pour les éléments de structure : coques multicouches, poutres multi-fibres, …).
Dans le cham_elem produit, chaque point (nœud ou point de Gauss) sera représenté par n sous-points (n étant déterminé par les choix de l'utilisateur dans la commande AFFE_CARA_ELEM).
Les n sous-points d'un point porteront les mêmes composantes (avec les mêmes valeurs).

Le mot clé CARA_ELEM est obligatoire : c'est le nom de la structure de données de type cara_elem qui permet de déterminer le nombre des sous-points.

3.1.7 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.

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

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.

Dans ce dernier cas, la numérotation fournie (objet de type PROF_CHNO) est dupliquée afin que le champ créé soit autoportant.

Si des Lagrange issus de conditions aux limites dualisées sont présents, une vérification est faite par l'opérateur. En effet, si les champs considérés ne satisfont pas les mêmes conditions limites, ils ne sont pas compatibles. Une alarme est alors émise : les Lagrange seront mis à zéro dans le champ résultant.

3.1.8 Opérande TITRE = titr

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

3.1.9 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 aux nœuds 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.

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

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 et GROUP_NO

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

Si TYPE_CHAM='NOEU_xxx', on affecte des nœuds ; l'utilisation de l'opérande GROUP_MA 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 de l'opérande GROUP_NO est alors interdite.

Remarque :
Pour le mot clé facteur AFFE, le mot clé TOUT='OUI', veut dire :
- "tous les nœuds du maillage" pour les cham_no,
- "toutes les mailles du maillage" pour les cartes,
- "tous les éléments du modèle" pour les cham_elem,

3.3.3 Opérande NOM_CMP

Les noms des composantes que l'on veut affecter sont donnés par l'opérande NOM_CMP.
Si la grandeur est 'VARI_R', les composantes doivent être nommées 'V1', 'v2', 'V3', ...
Si la grandeur est 'VARI_R', et que l'on choisit PROL_ZERO='OUI', les composantes dont le numéro est inférieur au plus grand numéro affecté sont affectées à zéro. Pour plus de détails voir le document [U2.01.09].

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= 3.5e-2),
          _F  (GROUP_NO=('GN5',’GN7’,’GN9’) , NOM_CMP= 'DY' , VALE= 1.6e-2),
                    )
)
```

Création d'une carte de température (fonctions) :

```
TEMPF = CREA_CHAMP( OPERATION= 'AFFE',
```


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

```
            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.
- une carte en assemblant des morceaux de cartes et/ou de cham_elem/ELEM.

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.

3.4.2 Opérandes MAILLAGE, MODELE

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 et GROUP_NO

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 l'opérandes GROUP_NO.
Le mot clé TOUT='OUI' veut dire ici : « tous les nœuds, mailles ou éléments qui portent des composantes (de la liste NOM_CMP) dans le champ ch1 ».

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',
```


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

```
        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 :

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

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 COMB 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

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

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 et il faut également qu'aucune composante d'un champ paramètre ne soit présente dans un autre champ paramètre de la liste.

Remarque : Ainsi pour les champs par éléments, il n'est par exemple pas possible de fournir deux champs de grandeur NEUT même si ce ne sont pas les mêmes composantes qui doivent être prises en compte sur les deux champs car chaque champ possède les composantes X1 à X30.

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'instants.

```
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 un cham_no/NEUT_R:
# ----------------------------------------------------------------------
CHN=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',)),
```


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

```
                    _F( TOUT = 'OUI', CHAM_GD = CH2,
                       NOM_CMP = ('TEMP',),  NOM_CMP_RESU = ('X2',)),)    )

# 2) multiplication   CH3N = CHN/X1 * CHN/X2 :
# ------------------------------------------
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) )
# On a choisi ici la composante X3, mais on pourrait choisir n'importe
quelle composante de X1 à X30.


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


# 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
CHAM_ELEM / ELEM    -> CHAM_ELEM / ELGA
CHAM_ELEM / ELEM    -> CHAM_ELEM / ELNO
```

Les ingrédients des traitements sont :

CARTE / ELEM -> 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 :

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

- 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 mot clé MODELE

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, et GROUP_MA 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 concourrant 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 5 possibilités supplémentaires :

- 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'.
- on peut extraire le champ d'abscisse curviligne d'un maillage. Il faut alors utiliser les mots clés : MAILLAGE = ma, NOM_CHAM = 'ABSC_CURV' et TYPE_CHAM = 'CART_ABSC_R'.
- 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.
- 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'

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

'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

- on peut extraire les différents champs contenus dans les structures de données cara_elem et char_meca. Cette fonctionnalité est plutôt réservée aux développeurs. Elle permet par exemple de tester le contenu de ces structures de données.

  Exemple :
```
CACOQU=CREA_CHAMP(  TYPE_CHAM='CART_CACOQU',OPERATION='EXTR',
                    CARA_ELEM=CARA,  NOM_CHAM='.CARCOQUE',)
```

  Les différents champs que l'on peut extraire sont donnés dans le tableau ci-dessous.

concept	NOM_CHAM	grandeur	Type du champ
cara_elem	'.CAFIBR'	CAFI_R	cham_elem
cara_elem	'.CANBSP'	NBSP_I	cham_elem
cara_elem	'.CARARCPO'	CAARPO	carte
cara_elem	'.CARCABLE'	CACABL	carte
cara_elem	'.CARCOQUE'	CACOQU	carte
cara_elem	'.CARDINFO'	CINFDI	carte
cara_elem	'.CARDISCA'	CADISA	carte
cara_elem	'.CARDISCK'	CADISK	carte
cara_elem	'.CARDISCM'	CADISM	carte
cara_elem	'.CARDNSCA'	CADISA	carte
cara_elem	'.CARDNSCK'	CADISK	carte
cara_elem	'.CARDNSCM'	CADISM	carte
cara_elem	'.CARGENBA'	CAGNBA	carte
cara_elem	'.CARGENPO'	CAGNPO	carte
cara_elem	'.CARGEOPO'	CAGEPO	carte
cara_elem	'.CARMASSI'	CAMASS	carte
cara_elem	'.CARORIEN'	CAORIE	carte
cara_elem	'.CARPOUFL'	CAPOUF	carte
char_meca	'.CVENTCXF'	VENTCX_F	carte
char_meca	'.CHME.EPSIN'	EPSI_R	carte
char_meca	'.CHME.F1D1D'	FORC_R	carte
char_meca	'.CHME.F1D2D'	FORC_R	carte
char_meca	'.CHME.F1D3D'	FORC_R	carte
char_meca	'.CHME.F2D2D'	FORC_R	carte

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

char_meca	'.CHME.F2D3D'	FORC_R	carte
char_meca	'.CHME.F3D3D'	FORC_R	carte
char_meca	'.CHME.FCO2D'	FORC_R	carte
char_meca	'.CHME.FCO3D'	FORC_R	carte
char_meca	'.CHME.FELEC'	FELECR	carte
char_meca	'.CHME.FL101'	FLAPLA	carte
char_meca	'.CHME.FL102'	FLAPLA	carte
char_meca	'.CHME.FLUX'	FTHM_R	carte
char_meca	'.CHME.FORNO'	FORC_R	carte
char_meca	'.CHME.IMPE'	IMPE_R	carte
char_meca	'.CHME.ONDE'	ONDE_R	carte
char_meca	'.CHME.ONDPL'	NEUT_R	carte
char_meca	'.CHME.ONDPR'	NEUT_R	carte
char_meca	'.CHME.PESAN'	PESA_R	carte
char_meca	'.CHME.PRESS'	PRES_R	carte
char_meca	'.CHME.ROTAT'	ROTA_R	carte
char_meca	'.CHME.SIGIN'	SIEF_R	carte
char_meca	'.CHME.SIINT'	NEUT_K8	carte
char_meca	'.CHME.VNOR'	SOUR_R	carte

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'	'ETOT_ELEM'	'ELEM_ENER_R'
'ACCE_ABSOLU'	'NOEU_DEPL_R'	'ETOT_ELGA'	'ELGA_ENER_R'
'COHE_ELEM'	'ELEM_NEUT_R'	'ETOT_ELNO'	'ELGA_ENER_R'
'COMPORTEMENT'	'CART_COMPOR'	'FERRAILLAGE'	'ELEM_FER2_R'
'COMPORTHER'	'CART_COMPOR'	'FLHN_ELGA'	'ELGA_FLHN_R'
'DEGE_ELGA'	'ELGA_EPSI_R'	'FLUX_ELGA'	'ELGA_FLUX_R'
'DEGE_ELNO'	'ELNO_EPSI_R'	'FLUX_ELNO'	'ELNO_FLUX_R'
'DEGE_NOEU'	'NOEU_EPSI_R'	'FLUX_NOEU'	'NOEU_FLUX_R'
'DEPL'	'NOEU_DEPL_R'	'FORC_AMOR'	'NOEU_DEPL_R'
'DEPL_ABSOLU'	'NOEU_DEPL_R'	'FORC_EXTE'	'NOEU_DEPL_R'
'DEPL_VIBR'	'NOEU_DEPL_R'	'FORC_LIAI'	'NOEU_DEPL_R'
'DERA_ELGA'	'ELGA_DERA_R'	'FORC_NODA'	'NOEU_DEPL_R'
'DERA_ELNO'	'ELNO_DERA_R'	'GEOMETRIE'	'NOEU_GEOM_R'
'DERA_NOEU'	'NOEU_DERA_R'	'HYDR_ELNO'	'ELNO_HYDR_R'
'DISS_ELEM'	'ELEM_DISS_R'	'HYDR_NOEU'	'NOEU_HYDR_R'
'DISS_ELGA'	'ELGA_DISS_R'	'INDC_ELEM'	'ELEM_NEUT_I'
'DISS_ELNO'	'ELNO_DISS_R'	'INDL_ELGA'	'ELGA_INDL_R'
'DISS_NOEU'	'NOEU_DISS_R'	'INTE_ELNO'	'ELNO_INTE_R'
'DIVU'	'NOEU_EPSI_R'	'INTE_NOEU'	'NOEU_INTE_R'
'DURT_ELNO'	'ELNO_DURT_R'	'IRRA'	'NOEU_IRRA_R'
'DURT_NOEU'	'NOEU_DURT_R	'META_ELNO'	'ELNO_VARI_R'
'ECIN_ELEM'	'ELEM_ENER_R'	'META_NOEU'	'NOEU_VARI_R'
'EFGE_ELGA'	'ELGA_SIEF_R'	'MODE_FLAMB'	'NOEU_DEPL_R'

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

'EFGE_ELNO'	'ELNO_SIEF_R'	'MODE_STAB'	'NOEU_DEPL_R'
'EFGE_NOEU'	'NOEU_SIEF_R'	'NEUT'	'NOEU_NEUT_R'
'ENDO_ELGA'	'ELGA_SIEF_R'	'PDIL_ELGA'	'ELGA_PDIL_R'
'ENDO_ELNO'	'NOEU_SIEF_R'	'PRAC_ELNO'	'ELNO_PRAC_R'
'ENDO_NOEU'	'ELNO_SIEF_R'	'PRAC_NOEU'	'NOEU_PRAC_R'
'ENEL_ELEM'	'ELEM_ENER_R'	'PRES'	'NOEU_PRES_C'
'ENEL_ELGA'	'ELGA_ENER_R'	'PRME_ELNO'	'ELNO_PRME_R'
'ENEL_ELNO'	'ELNO_ENER_R'	'PTOT'	'NOEU_DEPL_R'
'ENEL_NOEU'	'NOEU_ENER_R'	'QIRE_ELEM'	'ELEM_ERRE_R'
'EPEQ_ELGA'	'ELGA_EPSI_R'	'QIRE_ELNO'	'ELNO_ERRE_R'
'EPEQ_ELNO'	'ELNO_EPSI_R'	'QIRE_NOEU'	'NOEU_ERRE_R'
'EPEQ_NOEU'	'NOEU_EPSI_R'	'QIZ1_ELEM'	'ELEM_ERRE_R'
'EPFD_ELGA'	'ELGA_EPSI_R'	'QIZ2_ELEM'	'ELEM_ERRE_R'
'EPFD_ELNO'	'ELNO_EPSI_R'	'REAC_NODA'	'NOEU_DEPL_R'
'EPFD_NOEU'	'NOEU_EPSI_R'	'SECO_ELEM'	'ELEM_NEUT_R'
'EPFP_ELGA'	'ELGA_EPSI_R'	'SIEF_ELGA'	'ELGA_SIEF_R'
'EPFP_ELNO'	'ELNO_EPSI_R'	'SIEF_ELNO'	'ELNO_SIEF_R'
'EPFP_NOEU'	'NOEU_EPSI_R'	'SIEF_NOEU'	'NOEU_SIEF_R'
'EPME_ELGA'	'ELGA_EPSI_R'	'SIEQ_ELGA'	'ELGA_SIEF_R'
'EPME_ELNO'	'ELNO_EPSI_R'	'SIEQ_ELNO'	'ELNO_SIEF_R'
'EPMG_ELGA'	'ELGA_EPSI_R'	'SIEQ_NOEU'	'NOEU_SIEF_R'
'EPMG_ELNO'	'ELNO_EPSI_R'	'SIGM_ELGA'	'ELGA_SIEF_R'
'EPMG_NOEU'	'NOEU_EPSI_R'	'SIGM_ELNO'	'ELNO_SIEF_R'
'EPMQ_ELGA'	'ELGA_EPSI_R'	'SIGM_NOEU'	'NOEU_SIEF_R'
'EPMQ_ELNO'	'ELNO_EPSI_R'	'SING_ELEM'	'ELEM_SING_R'
'EPMQ_NOEU'	'NOEU_EPSI_R'	'SING_ELNO'	'ELNO_SING_R'
'EPOT_ELEM'	'ELEM_ENER_R'	'SIPM_ELNO'	'ELNO_SIEF_R'
'EPSA_ELNO'	'ELNO_EPSI_R'	'SIPO_ELNO'	'ELNO_SIEF_R'
'EPSA_NOEU'	'NOEU_EPSI_R'	'SIPO_NOEU'	'NOEU_SIEF_R'
'EPSG_ELGA'	'ELGA_EPSI_R'	'SIRO_ELEM'	'ELEM_SIEF_R'
'EPSG_ELNO'	'ELNO_EPSI_R'	'SISE_ELNO'	'ELNO_SIEF_R'
'EPSG_NOEU'	'NOEU_EPSI_R'	'SIZ1_NOEU'	'NOEU_SIEF_R'
'EPSI_ELGA'	'ELGA_EPSI_R'	'SIZ2_NOEU'	'NOEU_SIEF_R'
'EPSI_ELNO'	'ELNO_EPSI_R'	'SOUR_ELGA'	'ELGA_SOUR_R'
'EPSI_NOEU'	'NOEU_EPSI_R'	'STRX_ELGA'	'ELGA_STRX_R'
'EPSP_ELGA'	'ELGA_EPSI_R'	'TEMP'	'NOEU_TEMP_R'
'EPSP_ELNO'	'ELNO_EPSI_R'	'THETA'	'NOEU_DEPL_R'
'EPSP_NOEU'	'NOEU_EPSI_R'	'UTXX_ELGA'	Voir remarque ci-dessous
'EPVC_ELGA'	'ELGA_EPSI_R'	'UTXX_ELNO'	Voir remarque ci-dessous
'EPVC_ELNO'	'ELNO_EPSI_R'	'UTXX_NOEU'	Voir remarque ci-dessous
'EPVC_NOEU'	'NOEU_EPSI_R'	'VAEX_ELGA'	'ELGA_NEUT_R'
'ERME_ELEM'	'ELEM_ERRE_R'	'VAEX_ELNO'	'ELNO_NEUT_R'
'ERME_ELNO'	'ELNO_ERRE_R'	'VAEX_NOEU'	'NOEU_NEUT_R'
'ERME_NOEU'	'NOEU_ERRE_R'	'CONT_NOEU'	'NOEU_INFC_R'
'ERTH_ELEM'	'ELEM_ERRE_R'	'VARC_ELGA'	'ELGA_VARC_R'
'ERTH_ELNO'	'ELNO_ERRE_R'	'VARI_ELGA'	'ELGA_VARI_R'
'ERTH_NOEU'	'NOEU_ERRE_R'	'VARI_ELNO'	'ELNO_VARI_R'
'ERZ1_ELEM'	'ELEM_ERRE_R'	'VARI_NOEU'	'NOEU_VAR2_R'
'ERZ2_ELEM'	'ELEM_ERRE_R'	'VITE'	'NOEU_DEPL_R'
'ETHE_ELEM'	'ELEM_ENER_R'	'VITE_ABSOLU'	'NOEU_DEPL_R'

Remarques :

- Les champs « utilisateurs » nommés UT01_ELGA, …, UT19_ELNO peuvent être associés à des grandeurs différentes selon le contexte. Pour connaître le nom de la grandeur, on peut regarder le tableau écrit dans le fichier .mess en fin d'exécution associé à la structure de données résultat.

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

- 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. Le nom des mailles et des nœuds doivent être codés sur 8 caractères, les tables imprimées au format 'TABLEAU' impriment les chaînes sur 24 caractères par défaut, il est donc préférable d'utiliser le format 'ASTER' lors de la relecture des tables pour préciser le format des chaînes de caractères et d'indiquer K8 pour les paramètres NOEUD et MAILLE.

Les colonnes contenant les valeurs du champ (réels) doivent être identifiées par leur nom de composante dans la grandeur. Par exemple : DX, DY, DZ pour le déplacement (DEPL_R).

Les autres colonnes à renseigner dépendent du type de champ à créer :

Type du champ	Colonne 1	Colonne 2	Colonne 3
NOEUD_XXXX	NOEUD		
ELEM_XXXX	MAILLE	[SOUS-POINT]	
ELNO_XXXX	MAILLE	NOEUD	[SOUS-POINT]
ELGA_XXXX	MAILLE	POINT	[SOUS-POINT]

Le paramètre NOEUD contient le nom du nœud.
Le paramètre MAILLE contient le nom de la maille.
Le paramètre POINT contient le numéro du point de Gauss 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 nœud).

Attention : la table ne doit pas contenir d'autre colonne que les colonnes attendues (par exemple : NUME_ORDRE, INST, …). Si la table contient des colonnes inutiles, il faut les supprimer à l'aide de la commande CALC_TABLE + OPERATION='EXTR'.

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 Opérandes 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

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

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.

3.9.6 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).

3.9.6.1 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'	idem pour les minimums,
/ 'MINI_ABS'	
/ '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.

3.9.6.2 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. Il faut utiliser VALE_ABS pour avoir la valeur absolue.

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

Remarque :

Si par exemple :
- *NOM_CHAM = 'FLUX_ELNO',*
- *TYPE_RESU = 'INST',*

Le champ résultat est un champ de type FLUX_R qui contient des valeurs d'instants !

/ 'VALE_ABS' , disponible uniquement avec un TYPE_MAXI absolu ('MAXI_ABS' ou 'MINI_ABS'), le champ résultat contient alors les valeurs absolues extrêmes rencontrées au cours du transitoire.

3.9.6.3 Opérandes TOUT_ORDRE / LIST_INST / LIST_FREQ / PRECISION / CRITERE

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

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.

3.9.7 Exemples

3.9.7.1 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.

3.9.7.2 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 4ème mode propre du résultat modes (de type mode_meca).

3.9.7.3 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'     )
```

3.9.7.4 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')
```

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

```
nor_DNOR = CREA_CHAMP( TYPE_CHAM = 'NOEU_GEOM_R',
```


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

```
                                    OPERATION= 'NORMALE', MODELE= MO, GROUP_MA= 'FE'
      );
```


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