Operator CREA_CHAMP

1 Drank

To create a field of the cham_no type, card or cham_elem. One can create a field

- by assignment of values on nodes or meshes,
- by assembling (and/or combining) pieces of existing fields,
- by evaluating the functions of a field of functions to make a field of realities of it,
- by modifying the geometrical representation of a field (transition nodes ⇔ Gauss points for example),
- by extracting a field from a data structure of the type SD_RESULTAT (evol_ther, evol_noli, mode_meca,…).
- starting from the values contained in a data structure of the type COUNTS,
- …

One can also make use of this command to combine several fields linearly, but one can make "complicated" combinations more: multiplication of fields…

Produced a data structure of the cham_no type or card or cham_elem

Note:: The command is not D entering (reuse) except for operation "ASSE"

Contents

1 But1.....
2 Syntaxe4.....
3 Opérandes8.....
3.1 Operands Generals .....
8.3.1.1 Operand TYPE_CHAM, assignment of a type at the field result .....
8.3.1.2 Operand MAILLAGE = mailla9.....
3.1.3 Operand PROL_ZERO9.....
3.1.4 Operands MODELS, OPTION .....
9.3.1.5 Key words NUME_DDL and CHAM_NO9.....
3.1.6 Operand TITER = titr9.....
3.1.7 Operand INFO = /1 /210.....
3.2 Operand OPERATION =/“AFFE”/“ASSE”/“EVAL”/“DISC”/“EXTR”/“R2C”/“C2R”/“COMB'..... 10
3.3 Operands for OPERATION = “AFFE'..... 10
3.3.1 Key word factor AFFE10.....
3.3.2 Operands TOUT=' OUI', GROUP_MA, GROUP_NO, MESH and NOEUD10.....
3.3.3 Operand NOM_CMP10.....
3.3.4 Operands VALE, VALE_I, VALE_C or VALE_F11.....
3.3.5 Remarque11.....
3.3.6 Exemples11.....
3.4 Operands for OPERATION = “ASSE'..... 11
3.4.1 Généralités11.....
3.4.2 Operands MAILLAGE, MODELE .....
12.3.4.3 Operands for the key word factor ASSE12.....
3.4.4 Exemples13.....
3.5 Operands for the key word factor COMB13.....
3.5.1 Operand CHAM_GD = ch113.....
3.5.2 Operand COEF_R = coefr13.....
3.5.3 Exemple13.....
3.6 Operands for OPERATION = “EVAL'..... 14
3.6.1 Operand CHAM_F= chf14.....
3.6.2 Operand CHAM_PARA= l_chpara14.....
3.6.3 Exemples14.....
3.7 Operands for OPERATION = “DISC'..... 15
3.7.1 Operand CHAM_GD= ch116.....
3.7.2 key word MODELE .....
16.3.7.3 Exemple16.....
3.8 Operands for OPERATION = “NORMALE'..... 16
3.9 Operands for OPERATION = “EXTR'..... 16
3.9.1 Typing of the field result, key word TYPE_CHAM18.....

3.9.2 Operand TABLE20.....
3.9.3 Operand RESULTAT21.....
3.9.4 Operand NOM_CHAM21.....
3.9.5 Operands NUME_ORDRE/NUME_MODE/NOM_MODE/…/INTERPOL21.....
3.9.6 Computation of a field containing the “extrema” of a SD_RESULTAT21.....
3.9.7 Exemples22.....

2 Syntax

```
ch2 [*] = CREA_CHAMP    (

      ◊TITER  =titer      ,                                        [l_Kn]
      ◊INFO    =/1     ,                                          [DEFAULT]
                  /2  ,
      ♦TYPE_CHAM=/        "NOEU_xxx",
                         /"CART_xxx",
                         /"ELNO_xxx",
                         /"ELGA_xxx",
                         /"ELEM_xxx",

      # To impose the classification of field result
      # (only allowed for the cham_no):
      ◊/NUMÉRIQUE_DDL      = nu  ,
[nume_ddl]
           /CHAM_NO  = cnonu,                                  [cham_no]

      # If the "structure" of field result (cham_elem or cham_no)
      # requires values which one cannot determine:
      ◊PROL_ZERO=/              "NON",                          [DEFAULT]
                           /   "OUI',



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

         ♦/MAILLAGE=ma                 ,                        [mesh]

      .       /MODELE=mo                    ,                      [model]

         ♦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                    ,
[field]

         ♦/MAILLAGE=ma                    ,                        [mesh]
            /MODELE          =mo            ,
[model]
               ◊OPTION=option                                        [kN]

         ♦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   ,              [field]

                           ◊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)]
         will ♦CHAM_PARA=l_ch_para       ,                [l_champ]

      /OPERATION       =            ' DISC',
      #================================
         #si  the field result is a CHAM_ELEM:
         ◊MODELE=mo                    ,                          [model]
            ◊OPTION=          option                              [kN]

         ♦CHAM_GD=ch1              ,                              [field]

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

|MAILLE=l_mail , [l_maille]

```
        /OPERATION       =            ' R2C',
        #==================================
           # to transform a real field into complex field
           # (with imaginary part null)
           ♦CHAM_GD=chR                        ,                [field]

        /OPERATION       =            ' C2R',
        #==================================
           # to transform a field complexes in real field:
           ♦CHAM_GD=chC                  ,                      [field]
           ♦PARTIE=/                     "REEL",    # left real
                                 /   "IMAG",    # left imaginary
                                 /   "MODULE",  # absolute value
                                 /   "PHASE",   # "phase" (in degrees)


        /OPERATION       =            ' EXTR',
        #================================

           ♦#extraction     of the field of geometry of one mesh:
              /♦MAILLAGE       =ma      ,                          [mesh]
                 ♦NOM_CHAM  =   ' GEOMETRIE',

              #extraction   in an array:
              /♦      TABLE=tabl            ,                     [array]
                 ♦/MAILLAGE=ma            ,                     [mesh]
                    /MODELE       =mo     ,                        [model]
                       ◊OPTION=option           ,                 [kN]

              #extraction   of the "level set" of a SD fiss_xfem:
              /♦FISSURE     =fxfem      ,
[fiss_xfem]
                 ♦NOM_CHAM  =      "LTNO",
                                /   "LNNO",
                                /   "GRLTNO",
                                /   "GRLNNO",
                                /   "STNO",
                                /   "STNOR",
                                /   "BASLOC",

              #extraction   of a field of a SD cara_elem:
              /♦CARA_ELEM      =carele     ,
[cara_elem]
                 ♦NOM_CHAM  =nomch   ,                        [kN]

              #extraction   of a field of a SD char_meca:
              /♦CHARGE        =charge      ,
[char_meca]
                 ♦NOM_CHAM    =nomch ,                        [kN]
```

```
      #extraction   of a field of a SD_RESULTAT:
                  /♦RESULTAT        =RESU ,
                     ♦NOM_CHAM  =      "ACCE",
                                     /. . (cf [22]),

                     ♦/#Sélection         of a sequence number in the
      SD_RESULTAT
                           /NUME_ORDRE       =nuordr     ,              [I]
                           /NUME_MODE           =numode     ,              [I]
                           /NOEUD_CMP           =   (node, cmp),         [l_K8]
                           /NOM_CAS          =nocas   ,                  [kN]
                           /ANGLE            =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      of a field containing the "extrema" of
                           a #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]

                                 )

                        SiTYPE_CHAM='        NOEU_xxx'alors   [*]    =cham_no
                              "CART_xxx'                            card
                              "                          ELNO_xxx'cham_elem
                              "                          ELGA_xxx'cham_elem
                              "                          ELEM_xxx'cham_elem
```

3 Operands

3.1 Operands Generals

3.1.1 Operand TYPE_CHAM, assignment of a type at the field result

This key word (compulsory) is initially used to typify the field result of the command. It is made of 2 "keys" connected by a "underscore" (_):

TYPE_CHAM= "" where:

```
          "NOEU'          Field at nodes                    (cham_no)
            /  "CART'         constant Field by mesh              (card)
            /  "ELNO'         Field by elements with the nodes    (cham_elem)
            /  "ELGA'         Field by elements with Gauss points (cham_elem)
            /  "ELEM'         constant Field by element           (cham_elem)

GD=/            "DEPL_R'        displacement
            /  "forced        SIEF_R'
            /  "TEMP_R'       temperature
            /  "FLUX_R'       flux
            /  …
```

the type of the field result is deduced from information given by the user. For example:

```
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)
```

This key word is also used to specify (for the command) which must be the nature of the field wanted as a result. It is essential for operations "AFFE", "ASSE" and "DISC".

Examples:

```
  OPERATION= "AFFE"  + TYPE_CHAM= "  CART_DEPL_R'=> a card of DEPL_R.
  OPERATION= "ASSE"  + TYPE_CHAM= "  NOEU_EPSI_R'=> a cham_no of EPSI_R.
  OPERATION= "DISC"  + TYPE_CHAM= "  NOEU_SIEF_R'=> a cham_no of SIEF_R.
```

There are only two operations for which this key word is a stress useless (but compulsory!) for the user (OPERATION=' EVAL" and OPERATION=' EXTR") because for these two operations, the nature of the field result is imposed by the choice of the operation.

The information of key word TYPE_CHAM is (unfortunately) tiresome for l'OPERATION=' EXTR". He results from key word NOM_CHAM. The correspondence is given in [20].

Notice important

the possibility of creating cham_elem of any quantity is conditioned by the level of development (data-processing) of element types finished model. All is not yet possible; for example, to create a cham_elem of FLUX_R on a model containing elements DKT , it is necessary that this finite element envisaged to do it (what is not the case today).

One cannot give here the precise list of the quantities allowed for each type of finite element. One will be satisfied to say roughly:

- *for the isoparametric elements of mechanics, the quantities are allowed: GEOM_R , INST_R , NEUT_R , NEUT_F , EPSI_R , SIEF_R , VARI_R , DOMMAG and HYDR_R ,*
- *for the isoparametric elements of thermal, are allowed the quantities: GEOM_R , INST_R , NEUT_R and NEUT_F .*

3.1.2 Operand MAILLAGE = netted

When one creates a CHAM_NO or a CARD, it is necessary to specify in general on which mesh this field will be based. For that, one uses the key word MAILLAGE.

3.1.3 Operand PROL_ZERO

When a CHAM_ELEM is created, the values existing in the field are determined by the finite elements of the model. For example, a field SIGM_ELGA on a model 2D **must** contain 4 components SIXX, SIYY, SIZZ and SIXY.
If the construction of the field does not make it possible to calculate all the expected values, one is confronted with a problem. If key word PROL_ZERO is worth "OUI', the missing values will be put at zero.
If key word PROL_ZERO is worth "NON", the code will stop in error.

This problem also relates to (but more rarely) the CHAM_NO when one wants to impose the classification of their components (see key keys NUME_DDL and CHAM_NO).

3.1.4 Operands MODELS, OPTION

When a CHAM_ELEM is created, it is necessary to specify on which finite elements the field will be defined. For that, the MODEL key word is used.

To describe structure of a CHAM_ELEM, it is not enough to give (via the model) a kind of element for each mesh, because a kind of element can know several "forms" for a given quantity. To create the desired field, the user can use key word OPTION. If for example, he writes: X= CREA_CHAMP (… MODELE=mo, OPTION=' SIEF_ELGA",…) , the field created by CREA_CHAMP will have the same form that if it had been calculated by CALC_CHAMP / CONTRAINTE=' SIEF_ELGA". If the user does not employ key word OPTION, CREA_CHAMP will choose (if it can it) a form by default.

Note: for operation "ASSE", when cham_elem are assembled, it is better in general not to provide key word OPTION. The option which will be selected will be that of the field provided in the 1st occurrence of the key word factor ASSE if the quantity associated with this field is the same one as that of the field result.

3.1.5 Key words NUME_DDL and CHAM_NO

These two key words make it possible to impose a classification for the field result (if this one is a CHAM_NO). One for the field gives via these key words a "model" of classification result.

If one gives NUME_DDL= nu, one will take as classification that of nu. This possibility is valid only for the fields of displacements ("MECHANICAL" phenomenon) or for fields of temperature (phenomenon "THERMAL") or fields of acoustic pressure ("ACOUSTIC" phenomenon).

If CHAM_NO= chno is given, the numeration of chno will be taken.

Notice on the disk space used:

> *Sometimes these 2 keywords make it possible to save much core on the "Total" basis. When for example, one extracts from many cham_no of a SD RESULTAT , if one does not use one of these key keys, one duplicates the profile of the field for each one of them. If one uses one of these 2 keywords, all these fields will lean on the profile contained in chno (or nu).*

3.1.6 Operand TITER = titr

Title which one wants result to give to the field [U4.03.01].

3.1.7 Operand INFO = /1 /2

INFO = 1
No printing.

INFO = 2
Printing on the file "MESSAGE" of the field result.

3.2 Operand OPERATION =/"AFFE"/"ASSE"/"EVAL"/"DISC"/"EXTR"/"R2C"/"C2R"/"COMB"

This operand is used result to choose the "mode" of fabrication of the field. One can create a field:

- by assignment of values on nodes or meshes (OPERATION=' AFFE"),
- by assembling pieces of fields defined on pieces of meshes (OPERATION=' ASSE"),
- by modifying the geometrical representation (discretization) of a field (transition nodes <-> Gauss points for example) (OPERATION=' DISC"),
- by extracting a field from a SD of the type SD_RESULTAT (evol_ther, evol_noli, mode_meca,...) (OPERATION=' EXTR").
- by extracting from the numerical values D" an array whose columns have preset names: "MESH", "NOEUD"...
- by combining linearly fields (OPERATION=' ASSE"),
- in "combining" (multiplication, exponential,...) fields (OPERATION=' EVAL"),
- by evaluating the functions of a field of functions to make of it a field of realities (OPERATION=' EVAL"),
- by transforming a real field into field complexes (or reciprocally) (OPERATION=' R2C"/"C2R"),
- by making a linear combination of several cham_no having same classification (OPERATION=' COMB"). Unlike operation "ASSE", the field result will preserve the ddls of Lagrange associated with the dualisation with the boundary conditions.

3.3 Operands for OPERATION = "AFFE"

This operation makes it possible to affect values (real, whole, complex or function) on geometrical entities (nodes or meshes) of a mesh.

Quantity associated with the field east implicitly given by key word TYPE_CHAM (above).

3.3.1 Key word factor AFFE

the operands are gathered under the key word factor AFFE. This key word is répétable. The principle of overload is applied between the various occurrences of key word AFFE : if a geometrical entity is affected several times, the last assignment carries it.

3.3.2 Operands TOUT=' OUI', GROUP_MA, GROUP_NO, MESH and NOEUD

the geometrical entities that one wants to affect are given by the operands TOUT=' OUI', GROUP_MA, GROUP_NO, MESH and NOEUD.

If TYPE_CHAM=' NOEU_xxx', one affects nodes; the use of operands GROUP_MA and MESH is possible and means that one affects all the nodes of meshes specified.
If TYPE_CHAM: "EL. _xxx" (or "CART_xxx"), one affects meshes; the use of operands GROUP_NO and NOEUD is then prohibited.

3.3.3 Operand NOM_CMP

the names of the components that one wants to affect are given by operand NOM_CMP.
If the quantity is "VARI_R", the components must be named "V1", "v2", "V3",...
If the quantity is "VARI_R", and that one chooses PROL_ZERO=' OUI', the components whose number is lower than greatest affected number the are assigned to zero. For more details to see the document [U2.01.09].

3.3.4 Operands VALE, VALE_I, VALE_C or VALE_F

the values to be affected are given by operands VALE, VALE_I, VALE_C or VALE_F according to nature (reality, integer, complex, function (or formulates)) of the components quantity (DEPL_R : reality, DEPL_C : complex, TEMP_F : function/formula,….).

3.3.5 Notice

the rule of remanence (see U1.03.00) applies for the various components which one can affect.

3.3.6 Creation

examples of a field at nodes of displacement. One wants to impose the classification of the field (that of cnomod):

```
DEPL1 = CREA_CHAMP (OPERATION= "AFFE",
          TYPE_CHAM=' NOEU_DEPL_R', MAILLAGE = MY , 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   (NOEUD= ("N5", "N7", "N9"), NOM_CMP= "DY", VALE= 1.6e-2),
                    )
)
```

Creation of a card of temperature (functions):

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

3.4 Operands for OPERATION = "ASSE"

3.4.1 General information

This operator "assembles" "pieces of fields" to manufacture new. Each occurrence of key word ASSE defines a piece of field. One calls a piece of field, the restriction of an existing field (card / cham_no or cham_elem) on a set of geometrical entities (meshes or nodes) and on a set of components.

There is a principle of overload of the occurrences of key word ASSE if the pieces recover the ones the others.

Currently, one can manufacture:

- a cham_no by assembling pieces of cham_no.
- a cham_elem by assembling pieces of cham_elem and/or cards.
- a card by assembling pieces of cards and/or cham_elem/ELEM.

Operation "ASSE" also makes it possible to change the quantity associated with a field; for example to transform a strain field (EPSI_R) into stress field (SIEF_R). For that it is necessary to use key words NOM_CMP and NOM_CMP_RESU.

The assembly of the pieces of fields can be done by cumulating the pieces (key words CUMUL and COEF_R). That makes it possible to use this command to make linear combinations of CHAM_NO or CHAM_ELEM.

Note: the operation "ASSE" is the only one for which the field result can be "reentrant".

3.4.2 Operands MAILLAGE, MODELS

Even use that for OPERATION= "AFFE" [§3.3.1] and [§3.3.2].

3.4.3 Operands for the key word factor ASSE

Each occurrence of the key word factor ASSE makes it possible to define a piece of field which one result assembles in the field.

3.4.3.1 Operand CHAM_GD = ch1

ch1 is the field (existing) with which one wants to manufacture a piece of field.

3.4.3.2 Operands TOUT=' OUI', GROUP_MA, GROUP_NO, MESH and NOEUD

These operands are used to define the geometrical restriction of the field ch1. If ch1 is a CHAM_NO, one can use all these operands. If ch1 is a CHAM_ELEM (or a CARD), one cannot use operands GROUP_NO and NOEUD.

3.4.3.3 Operands NOM_CMP and NOM_CMP_RESU

operand NOM_CMP is used to define the components on which one wants to restrict the field ch1. If NOM_CMP is absent, all the components of ch1 are taken.

Operand NOM_CMP_RESU is used to re-elect (if it is wished) the components of ch1. If NOM_CMP_RESU is provided, NOM_CMP owes the being also and the two lists in correspondence must be of the same length.

Example 1: to transform a field of EPSI_R into field of VARI_R

```
CHVARI=CRÉA_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"),   ),)
```

Example 2: to permute cmps SIXX and SIYY of a field of SIEF_R

```
CHS2=CRÉA_CHAMP (OPERATION=' ASSE', TYPE=' NOEU_SIEF_R',
   MAILLAGE= MY,
   ASSE=_F (CHAM_GD=CHS1, TOUT=' OUI',
      NOM_CMP  = ("SIXX", "SIYY"),
      NOM_CMP_RESU= ("SIYY", "SIXX",),  ))
```

3.4.3.4 Operands CUMUL , COEF_R and COEF_C

operand CUMUL=' OUI' wants to say that the values of the occurrence concerned will be added with the possible already existing values.

If CUMUL=' NON', the affected value replaces the value possibly already present (CUMUL=' OUI' is invalid for the fields of "text" (k8/k16,…) of course).

The operand COEF_R = coefr allows the multiplication of the piece of field by the real coefficient coefr before result assembling it at the field.

Example:

To manufacture the cham_elem : ch3 = 2. * ch1 + 3. * ch2

```
CH3= CREA_CHAMP (OPERATION= "ASSE",
          MODELS = 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.),)
 )
```

Notices concerning the complex fields

key word COEF_C is accepted only if the field result (CH3) and the fields arguments (CH1 and CH2) are all complex. To make a linear combination with complex coefficients of real fields, it is necessary to transform as a preliminary the real fields into complex fields. See OPERATION = "R2C".

3.4.4 Examples

Example 1

To manufacture a cham_no of temperature by extracting a field already calculated (in a evol_ther) and by redefining it (with 25. degrees) on the soudur1 mesh group.

```
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 = MY, TYPE_CHAM = ' NOEU_TEMP_R',
   ASSE =  (_F (CHAM_GD = CH1, TOUT = ' OUI',),
                 _F (CHAM_GD = CH2, GROUP_MA = SOUDUR1),)
)
```

Example 2:

To manufacture a cham_elem of VARI_R (to use it as an initial state for STAT_NON_LINE) by recovering the local variables (6 and 8) of a constitutive law to make of them variables 1 and 2 of (new) the constitutive law which will be used in the STAT_NON_LINE to come.

```
CH1= CREA_CHAMP (OPERATION= "EXTR", TYPE_CHAM=' ELGA_VARI_R',
          RESULTAT= STNL,   NOM_CHAM= "VARI_ELGA", INST = 4.)

CH2= CREA_CHAMP (OPERATION= "ASSE",
          MODELS = MO   ,    TYPE_CHAM = ' ELGA_VARI_R',
          ASSE = _F (CHAM_GD = CH1, TOUT = ' OUI',
                     NOM_CMP  =    ("V6", "V8"),
                     NOM_CMP_RESU = ("V1", "V2"),  ))
```

3.5 Operands for the key word factor COMB

This key word makes it possible to calculate the linear combination of several cham_no having same classification. Unlike operation "ASSE", the field result will also contain the coefficients of Lagrange corresponding to the dualisation of the boundary conditions.

Each occurrence of the key word factor ASSE makes it possible to define an element of the linear combination.

3.5.1 Operand CHAM_GD = ch1

ch1 is the cham_no (existing) that one wants to combine linearly.

3.5.2 Operand COEF_R = coefr

coefr is the real coefficient applied to ch1 for the combination.

3.5.3 Example

To compute: C = 1.*A - 2.*B, one writes:

```
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 Operands for OPERATION = "EVAL"

This operation is used to transform a field of functions into fields of realities by evaluating the functions of the field of functions.

The field of functions is obligatorily a field of quantity "NEUT_F" and the field result will be always a field of "NEUT_R". This field could be transformed into field of another unspecified quantity by calling one second time on command CREA_CHAMP/OPERATION=' ASSE'.

An example of the use of operation "EVAL" is given in the document [U2.01.09] "analytical Definition of a stress field…"

3.6.1 Operand CHAM_F= chf

chf is the name of the field of functions to be evaluated (CHAM_NO, CARD or CHAM_ELEM).

3.6.2 Operand CHAM_PARA= will l_chpara

will l_chpara is the list of the fields "parameters" for the evaluating of the functions. All the fields of will l_chpara must be discretized in the same way that chf. For example, if chf is a CHAM_ELEM/ELGA, it is necessary that all the fields of will l_chpara are also CHAM_ELEM/ELGA.

It is necessary that the list of the fields parameters is sufficient to allow the evaluating of all the functions referred in chf.

3.6.3 Examples

3.6.3.1 Example 1

One wants to create a cham_elem (SIEF_R) with Gauss points whose components are analytical functions of the geometry and time. For this example, it is supposed that one already manufactured two fields with Gauss points the CHGEOMG : field of geometry and CHINSTG : field of times.

```
RHO=1000.
G=10.
KP=3.
SIZZ = FORMULA (NOM_PARA = "Z", VALE = "RHO*G*Z")
SIXX = FORMULA (NOM_PARA = ("Z", "INST"), VALE = "KP*SIZZ (Z) +3.*INST")

# Assignment of the functions:
```

```
# ------------------------------------------------
SIG1=CRÉA_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")))

# Evaluating of the functions:
# ------------------------------------------------
SIG2= CREA_CHAMP (OPERATION=' EVAL', TYPE_CHAM=' ELGA_NEUT_R',
      MODELE=MO, CHAM_F=SIG1, CHAM_PARA= (CHGEOMG, CHINSTG)   )

 # transformation of the field of NEUT_R in SIEF_R:
 # ------------------------------------------------
SIG3=CRÉA_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 Example 2

One wants to calculate a field of temperature to the nodes ($CH3$) containing the product of 2 other fields at nodes of temperature ($CH1$ and $CH2$)

```
# 1) transformation of cham_no/TEMP_R (CH1 and CH2) in cham_no/NEUT_R:
# ---------------------------------------------------------------------
CH1N=CRÉA_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=CRÉA_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 = FORMULA (NOM_PARA = ("X1", "X2"), VALE = "X1*X2")

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


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


# 3) transformation of cham_no/NEUT_R (CH3N) in cham_no/TEMP_R:
# ---------------------------------------------------------------------
CH3=CRÉA_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 Operands for OPERATION = "DISC"

This operation is used to modify the “discretization” of an existing field. For example, to transform a field at nodes into fields with Gauss points.

The two fields (given and result) are associated with the same quantity. There exists an exception: the case of the operation DISC which makes it possible to transform a cham_no_VAR2_R (or a carte_VAR2_R) into cham_elem_VARI_R (see for example the test hplp100b).

The discretization wanted by the user for his field result is indicated by key word TYPE_CHAM.

Only the following cases are treated currently by the command:

CARD	- > CHAM_ELEM/ELNO
CARD	- > CHAM_ELEM/ELGA
CARD	- > CHAM_ELEM/ELEM
CARD	- > 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

the ingredients of the processing are:

CARD - > ELxx:

- The value (single) carried by a mesh is recopied on all the points of the mesh.

NOEU - > ELxx:

- the transition of the values of the nodes at the intern points of the mesh is made the shape functions by means of the finite elements model.

ELGA - > ELNO:

- the transition of the values of the intern points to the nodes of the mesh is made the matrix of Gauss->Nœuds extrapolation by means of.

ELNO - > NOEU:

- the transition with the values of the nodes of the mesh is done by arithmetic mean of the values carried by the nodes of the convergent elements.

3.7.1 Operand CHAM_GD= ch1

ch1 is the field which one wants to modify the “discretization”.

3.7.2 key word MODELS

Even use that for OPERATION = “AFFE” [3.3.2]

3.7.3 Example

```
# CHXG = FIELD OF GEOMETRIE TO Gauss points:
# ------------------------------------------------------
CHXN =CRÉA_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 Operands for OPERATION = "NORMAL"

This operation is used to calculate the "normal" vectors with the facets of a model. The user must indicate with the key keys MODELS, GROUP_MA and MESH the name of the model concerned as well as the "facets" of which it wishes to calculate the norms. The "facets" can be elements of "skin" of a mesh 3D or shell elements/shell.
For the meshes 2D, the "facets" are linear elements.

The produced field is a cham_no (quantity GEOM_R) whose components are named X, Y, Z. the norm carried by a node is obtained by realising the norms of the facets concourrant in this node. The "normal" vector is length 1.

3.9 Operands for OPERATION = "EXTR"

This operation is in general used to extract a field from a SD as resultat_sdaster type.

There are 4 additional possibilities:

1) one can extract the field from geometry of the nodes of a mesh. The keywords should then be used: MAILLAGE = my, NOM_CHAM = "GEOMETRIE" and TYPE_CHAM = "NOEU_GEOM_R".
2) one can create a field by extracting from an array the values corresponding to parameters of pre-established names: NET, NOEUD, POINT, SOUS_POINT, names of the components.
3) one can extract the "level set" associated with a crack XFEM. The keywords should then be used:
   FISSURE = fiss_xfem, NOM_CHAM = "LTNO" /"LNNO"
   /"GRLTNO" /"GRLNNO"
   /"STNO" /"STNOR"
   /"BASLOC"

   and TYPE_CHAM = "NOEU_NEUT_R" except "STNO" for which it will be tangent "NOEU_NEUT_I

   " " LTNO' level normal
   set " LNNO' level set
   "GRLTNO' gradient of the level set tangent
   "GRLNNO' gradient of the level set norm
   "STNO' statute of the nodes, 1 whole value
   "STNOR' statute of the nodes, 1 actual value
   "BASLOC' bases local in crack tip, with 6/9 actual values, 2/3 coordinates for the point and 2/3 coordinates for each vector of the base local in 2D/3D

4) one can extract the various fields contained in the data structures cara_elem and char_meca. This functionality is rather reserved for the developers. It makes it possible for example to test the contents of these data structures.

   Example:
   CACOQU=CRÉA_CHAMP (TYPE_CHAM=' CART_CACOQU", OPERATION=' EXTR",
   CARA_ELEM=CARA, NOM_CHAM=" .CARCOQUE",)

   the various fields which one can extract are given in the table below.

concept	NOM_CHAM	standard size	of the field
cara_elem	" .CAFIBR"	CAFI_R	cham_elem

cara_elem	” .CANBSP'	NBSP_I	cham_elem
cara_elem	“.CARARCPO”	CAARPO	card
cara_elem	“.CARCABLE”	CACABL	card
cara_elem	“.CARCOQUE”	CACOQU	card
cara_elem	“.CARDINFO”	CINFDI	card
cara_elem	“.CARDISCA”	CADISA	card
cara_elem	“.CARDISCK”	CADISK	card
cara_elem	“.CARDISCM”	CADISM	card
cara_elem	“.CARDNSCA”	CADISA	card
cara_elem	“.CARDNSCK”	CADISK	card
cara_elem	“.CARDNSCM”	CADISM	card
cara_elem	“.CARGENBA”	CAGNBA	card
cara_elem	“.CARGENPO”	CAGNPO	card
cara_elem	“.CARGEOBA”	CAGEBA	card
cara_elem	“.CARGEOPO”	CAGEPO	card
cara_elem	“.CARMASSI”	CAMASS	card
cara_elem	“.CARORIEN”	CAORIE	card
cara_elem	“.CARPOUFL”	CAPOUF	card
char_meca	“.CVENTCXF”	VENTCX_F	card
char_meca	“.CHME.EPSIN”	EPSI_R	card
char_meca	“.CHME.F1D1D”	FORC_R	card
char_meca	“.CHME.F1D2D”	FORC_R	card
char_meca	“.CHME.F1D3D”	FORC_R	card
char_meca	“.CHME.F2D2D”	FORC_R	card
char_meca	“.CHME.F2D3D”	FORC_R	card
char_meca	“.CHME.F3D3D”	FORC_R	card
char_meca	“.CHME.FCO2D”	FORC_R	card
char_meca	“.CHME.FCO3D”	FORC_R	card
char_meca	“.CHME.FELEC”	FELECR	card
char_meca	“.CHME.FL101”	FLAPLA	card
char_meca	“.CHME.FL102”	FLAPLA	card
char_meca	“.CHME.FLUX”	FTHM_R	card
char_meca	“.CHME.FORNO”	FORC_R	card
char_meca	“.CHME.IMPE”	IMPE_R	card
char_meca	“.CHME.ONDE”	ONDE_R	card
char_meca	“.CHME.ONDPL”	NEUT_R	card
char_meca	“.CHME.ONDPR”	NEUT_R	card
char_meca	“.CHME.PESAN”	PESA_R	card
char_meca	“.CHME.PRESS”	PRES_R	card

char_meca	".CHME.ROTAT"	ROTA_R	card
char_meca	".CHME.SIGIN"	SIEF_R	card
char_meca	".CHME.SIINT"	NEUT_K8	card
char_meca	".CHME.VNOR"	SOUR_R	card

3.9.1 Typing of the field result, key word TYPE_CHAM

key word TYPE_CHAM (compulsory) [§3.2] must be well informed. Put except for the case of the extraction in a SD of the type DYNA_HARMO, key word TYPE_CHAM results from the symbolic name of the extracted field (NOM_CHAM). The table below gives the correspondence between these two keywords.

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"
"BEHAVIOR"	"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"
"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"	"NEAR"	"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"	See notices below
"EPVC_ELGA"	"ELGA_EPSI_R"	"UTXX_ELNO"	See notices below
"EPVC_ELNO"	"ELNO_EPSI_R"	"UTXX_NOEU"	See notices below
"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"	"VALE_CONT"	"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"	"QUICKLY"	"NOEU_DEPL_R"
"ETHE_ELEM"	"ELEM_ENER_R"	"VITE_ABSOLU"	"NOEU_DEPL_R"

Note:
• The fields "users" named UT01_ELGA,..., UT19_ELNO can be associated with different quantities according to the context. To know the name of the quantity, one can result look at the table written in the file .mess at the end of the execution associated with data structure.

• For the DYNA_HARMO, NOM_CHAM can take three values: "DEPL", "QUICKLY" and "ACCE". In the three cases, the type of the field result is a CHAM_NO/DEPL_C and it is thus necessary to inform: TYPE_CHAM=' NOEU_DEPL_C".

3.9.2 Operand COUNTS

♦TABLE = tabl

Name of concept counts containing the values to be stored in the field. The names of the parameters of the array must comply with certain rules.

The columns containing the values of the field (realities) must be identified by their name of component in the quantity. For example: DX, DY, DZ for displacement (DEPL_R).

The other columns with informing depend on the type of field to create:

Type of the field	Column 1	Column 2	Column 3
	NOEUD		
	NETS	[SUBPOINT]	
	NETS	[	SUBPOINT]
NOEUD	POINT	NETS	[SUBPOINT]

the parameter NOEUD contains the name of the node.
The parameter MESH contains the name of the mesh.
The parameter POINT contains the number of the Gauss point in the mesh.
Parameter SOUS_POINT (necessary only for the fields to "subpoints") contains the number of the subpoint in the Gauss point (or the node).

Caution: the array should not contain of another column only the expected columns (for example: NUME_ORDRE, INST,…). If the array contains useless columns, they should be removed using the command CALC_TABLE + OPERATION=' EXTR'.

3.9.3 Operand RESULTAT

♦RESULTAT = resu

Name of result concept in which one wants to recover a field.

3.9.4 Operand NOM_CHAM

♦NOM_CHAM

This key word specifies the symbolic name of the field to be extracted [U4.71.00].

3.9.5 Operands NUME_ORDRE/NUME_MODE/NOM_MODE/…/INTERPOL

These keys key are used to specify which is the sequence number of the SD_RESULTAT which one wants to extract.

The choice of the keywords to be used depends on the type of SD_RESULTAT [U4.71.00].

Remarks

When one uses INTERPOL = "LIN" , the extracted field will be an interpolation between two fields of the SD_RESULTAT . This interpolation always does not have a "physical" meaning; for example on eigen modes. This key word should be used only for the SD_RESULTAT of the type" evol_xxxx "
When one uses an access of the "real" type (INST or FREQ), one seeks a field in a given interval. If one finds several fields in the interval, the program stops in fatal error.

3.9.6 Computation of a field containing the "extrema" of a SD_RESULTAT

the idea is to create a field containing in each point of space the extreme value recorded during a transient (or time to which this value was recorded).

Today this paragraph relates to only the results of the evol_ther type, evol_elas, evol_noli and dyna_trans. The fields are always of "real" type.

One must specify:

- sequence numbers defining the transient: TOUT_ORDRE, LIST_INST,
- the selected type of extremality: TYPE_MAXI = "MAXI", …, "MINI_ABS", "NORM_TRAN",
- which one wants: the extreme value or time when this value is reached (TYPE_RESU).

3.9.6.1 Operand TYPE_MAXI

/"MAXI"	one considers the maximum reached by the components during the transient,
/"MAXI_ABS"	one of the components considers the maximum reached by the absolute value during the transient,
/"MINI"/"MINI_ABS"	idem for the minima,
/"NORM_TRAN"	one considers the maximum reached by the quantity: DX ** 2 + DY ** 2 + DZ ** 2 .

- for the 4 values: "MAXI", ..., "MINI_ABS", the components of the field are treated independently from/to each other: the extreme values can not be reached at the same time,
- the fifth value possible: "NORM_TRAN" is possible only if for the fields of depl_R. In a given point, one seeks time when the norm of vector translation is maximum and one result recopies in the field all the components of the field at found time.

3.9.6.2 Operand TYPE_RESU

/"VALE", the field result contains the extreme values recorded during the transient,

Note:

Even if the extremum were obtained with an absolute value ("MAXI_ABS" or "MINI_ABS"), the stored value is algebraic.

/"INST", the field result contains the values of times when the extreme values were recorded.

Caution: if for example:

```
NOM_CHAM  = "FLUX_ELNO",
TYPE_RESU = "INST",
```

the field result is a cham_no/FLUX_R which contains values of times!

3.9.6.3 Operands TOUT_ORDRE / LIST_INST / LIST_FREQ / accuracy / CRITERE

These keys key make it possible to specify the extent of the transient to be examined.

If TOUT_ORDRE = "OUI' one reviews all the sequence numbers.
If LIST_INST = linst one considers only specified times.

3.9.7 Extraction

3.9.7.1 examples of a field of temperature of a result concept of the type evol_ther

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

temp10 is the field of temperature extracted result the evoth (of evol_ther type) at time 10.

3.9.7.2 Extraction of a field of displacement of a result concept of the type mode_meca

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

mode4 is the 4th eigen mode of result the modes (of mode_meca type).

3.9.7.3 Extraction of the field of "temperature" containing times when the maximum temperature was reached during a transient

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

3.9.7.4 Extraction of a stress field in an array

Is the file (fort.81) containing the image of the following array:

```
FORCED
#DEBUT_TABLE #TITER "ELNO"
NETS  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
```

One can extract a field from "stresses" to the nodes of this array while making:

```
# reading of the array:
TA=LIRE_TABLE (UNITE=81, TYPE_TABLE=' TABLE', SEPARATEUR=' ")

# extraction of the field in the array:
CH=CRÉA_CHAMP (OPERATION=' EXTR", TYPE_CHAM=' ELNO_SIEF_R', TABLE=' TA',
               MODELE=MO, PROL_ZERO=' OUI', OPTION=' SIEF_ELNO_DEPL')
```

3.9.7.5 Computation of the field of the "norms" on a mesh group of edge

```
nor_DNOR = CREA_CHAMP (TYPE_CHAM = "NOEU_GEOM_R",
                       "NORMAL" OPERATION=, MODELE= MO, GROUP_MA= "FE"   );
```