
Opérateur STAT_NON_LINE

1 But

Calculer l'évolution mécanique ou thermo-hydro-mécanique couplée, en quasi-statique, d'une structure en non linéaire.

La non linéarité est liée soit au comportement du matériau (par exemple plastique), soit à la géométrie (par exemple en grands déplacements) soit au contact-frottement. Pour avoir des détails sur la méthode de résolution employée, on se reportera à la documentation de référence [R5.03.01].

L'évolution peut être étudiée en plusieurs travaux successifs (concept réentrant), soit en poursuite (le dernier instant calculé est l'instant initial du calcul suivant), soit en reprise en partant d'un instant antérieur.

Si le temps nécessaire pour effectuer le calcul n'est pas suffisant, le programme s'interrompt, mais les résultats déjà calculés sont sauvegardés si une base de données a été définie dans le profil d'étude de l'utilisateur. Produit une structure de données de type `evol_noli`.

Table des matières

1 But.....	1
2 Syntaxe.....	5
3 Opérandes.....	11
3.1 Opérandes MODELE / CHAM_MATER / CARA_ELEM	11
3.2 Mot clé EXCIT.....	11
3.2.1 Opérandes CHARGE	11
3.2.2 Opérande FONC_MULT.....	12
3.2.3 Opérande TYPE_CHARGE.....	12
3.3 Mot clé CONTACT.....	12
3.4 Mot-clé SOUS_STRUC.....	13
3.4.1 Opérande CAS_CHARGE.....	13
3.4.2 Opérandes TOUT / SUPER_MAILLE.....	13
3.4.3 Opérande FONC_MULT.....	13
3.5 Mots-clés COMP_INCR et COMP_ELAS.....	13
3.6 Mot clé ETAT_INIT.....	13
3.6.1 Opérandes SIGM/VARI/DEPL.....	14
3.6.2 Opérandes EVOL_NOLI.....	14
3.6.3 Opérande NUME_ORDRE / INST / NUME_DIDI.....	14
3.6.4 Opérande INST_ETAT_INIT.....	14
3.6.5 Opérande PRECISION / CRITERE.....	15
3.7 Mot clé INCREMENT.....	16
3.7.1 Opérande LIST_INST	16
3.7.2 Opérandes NUME_INST_INIT / INST_INIT / NUME_INST_FIN / INST_FIN.....	16
3.7.3 Opérande PRECISION.....	17
3.7.4 Opérande ERRE_TEMPS.....	18
3.8 Opérande METHODE.....	18
3.9 Mot clé NEWTON.....	18
3.9.1 Opérande PREDICTION.....	18
3.9.2 Opérande MATRICE.....	19
3.9.3 Opérande EVOL_NOLI.....	20
3.10 Mot clé RECH_LINEAIRE.....	20
3.10.1 Opérande METHODE.....	20
3.10.2 Opérande RESI_LINE_RELA / ITER_LINE_MAXI.....	21
3.10.3 Opérande PAS_MINI_CRIT / ITER_LINE_CRIT.....	21
3.10.4 Opérandes RHO_MIN/RHO_MAX/RHO_EXCL.....	21
3.11 Mot clé PILOTAGE.....	22
3.11.1 Opérande TYPE.....	22
3.11.2 Opérandes NOEUD/GROUP_NO	24
3.11.3 Opérandes TOUT/MAILLE/GROUP_MA.....	25

3.11.4 Opérande NOM_CMP.....	25
3.11.5 Opérande DIRE_PILO.....	25
3.11.6 Opérande FISSURE.....	25
3.11.7 Opérande COEF_MULT.....	25
3.11.8 Opérande ETA_PILO_R_MAX / ETA_PILO_R_MIN	26
3.11.9 Opérande ETA_PILO_MAX/ETA_PILO_MIN	26
3.11.10 Opérande PROJ_BORNES.....	27
3.11.11 Opérande SELECTION.....	27
3.12 Mot clé SOLVEUR.....	28
3.13 Mot clé CONVERGENCE.....	28
3.13.1 Opérande RESI_GLOB_RELA/RESI_GLOB_MAXI	28
3.13.2 Opérande RESI_COMP_RELA	28
3.13.3 Opérande RESI_REFE_RELA	29
3.13.4 Opérande ITER_GLOB_MAXI	29
3.13.5 Opérande ITER_GLOB_ELAS	29
3.13.6 Opérandes TYPE/PLATEAU_ITER/PLATEAU_RELA.....	30
3.13.7 Opérande ARRET.....	30
3.14 Mot-clé CRIT_FLAMB.....	30
3.14.1 Opérande LIST_INST / INST / PAS_CALC.....	31
3.14.2 Opérande PRECISION/CRITERE.....	31
3.14.3 Opérande CHAR_CRIT	31
3.14.4 Opérande NB_FREQ	31
3.14.5 Opérande RIGI_GEOM	31
3.14.6 Opérande DDL_EXCLUS	31
3.15 Mot-clé SENSIBILITE.....	32
3.16 Mot clé ARCHIVAGE.....	32
3.16.1 Opérande LIST_INST / INST / PAS_ARCH.....	32
3.16.2 Opérande PRECISION/CRITERE.....	32
3.16.3 Opérande DETR_NUME_SUIV.....	32
3.16.4 Opérande CHAM_EXCLU.....	33
3.17 Mot clé AFFICHAGE.....	33
3.17.1 Opérande UNITE.....	33
3.17.2 Opérande INFO_RESIDU.....	33
3.17.3 Opérande INFO_TEMPS.....	33
3.18 Mot clé OBSERVATION.....	33
3.18.1 Opérandes LIST_INST / INST / PAS_OBSE.....	34
3.18.2 Opérandes PRECISION / CRITERE.....	34
3.18.3 Opérandes NOM_CHAM / NOM_CMP.....	34
3.18.4 Opérandes TOUT/NOEUD/GROUP_NOEUD/MAILLE/GROUP_MA.....	34
3.18.5 Observation d'un champ ELGA	35
3.18.6 Observation d'un champ NOEU	36
3.19 Mot clé SUIVI_DDL.....	36

3.20 Contenu de la structure de données EVOL_NOLI.....	36
3.21 Opérande INFO.....	37
3.22 Opérande TITRE.....	37

2 Syntaxe

```

statnl[evol_noli] = STAT_NON_LINE
(
  reuse              =  statnl,                      [evol_noli]

♦   MODELE            =  mo,                          [modele]

♦   CHAM_MATER        =  chmat,                      [cham_mater]

◇   CARA_ELEM         =  carac,                      [cara_elem]

♦   EXCIT              =  _F(
  ♦   CHARGE           =  chi,                      [char_meca]
    FONC_MULT          =  fi,                      [fonction/formule]
    TYPE_CHARGE        =  /'FIXE_CSTE'             [DEFAULT]
                      /'FIXE_PILLO'
                      /'SUIV'
                      /'DIDI'
                      ),

◇   CONTACT           =  contact,                  [char_contact]

◇   SOUS_STRUC         =  _F(
  ♦   CAS_CHARGE       =  chi,                      [char_meca]
  ♦   /TOUT            =  'OUI',                  [DEFAULT]
    /SUPER_MAILLE     =  lma,                      [1_maille]
    FONC_MULT         =  fmult,
                      ),

♦   |COMP_INCR         =  _F(voir le document [U4.51.11]),
  |COMP_ELAS          =  _F(voir le document [U4.51.11]),

◇   ETAT_INIT          =  _F(
  ♦   /|SIGM           =  sig,                      [cham_elem_SIEF_R]
    |VARI              =  vain,                    [cham_elem_VARI_R]
    |DEPL              =  depl,                    [cham_no_DEPL_R]
    |STRX              =  strx                     [cham_elem_STRX_R]
    /EVOL_NOLI         =  evol,                    [evol_noli]
    /NUME_ORDRE        =  nuini,                    [I]
    /INST              =  instini,                  [R]
    PRECISION          =  /1.0E-3,                  [DEFAULT]
                      /prec,                        [R]
    CRITERE            =  /'RELATIF',                [DEFAULT]
                      /'ABSOLU',
    NUME_DIDI          =  nudidi,                    [I]
    INST_ETAT_INIT     =  istetaini                  [R]
                      ),

♦   INCREMENT          =  _F(
  ♦   LIST_INST        =  /litpsr8,                [litr8]
                      /litps,                      [list_inst]

    /NUME_INST_INIT    =  nuini,                    [I]
    /INST_INIT         =  instini,                  [R]
    /NUME_INST_FIN     =  nufin,                    [I]
    /INST_FIN          =  instfin,                  [R]
    PRECISION          =  /1.0E-3,                  [DEFAULT]
                      /prec,                        [R]
    /ERRE_TEMPS        =  /'NON'                    [DEFAULT]

```

```

                                =/'OUI'
                                ),

◇ METHODE                      =   /'NEWTON',                      [DEFAULT]
                                /'IMPL_EX',

◇ NEWTON                      =   _F(
    PREDICTION                  =   /'TANGENTE',                      [DEFAULT]
                                /'ELASTIQUE',
                                /'EXTRAPOL',
                                /'DEPL_CALCULE',
    EVOL_NOLI                    =   evol_noli,                      [evol_noli]
    MATRICE                      =   /'TANGENTE',                      [DEFAULT]
                                /'ELASTIQUE'
    REAC_INCR                    =   /1,                      [DEFAULT]
                                /mf,                      [I]
    REAC_ITER                    =   /0,                      [DEFAULT]
                                /it,                      [I]
    REAC_ITER_ELAS               =   /0,                      [DEFAULT]
                                /it,                      [I]
    PAS_MINI_ELAS               =   /0,                      [DEFAULT]
                                /pasmini,                      [R]
                                ),

◇ IMPL_EX                      =   _F(
    PREDICTION                  =   /'TANGENTE',                      [DEFAULT]
    REAC_INCR                    =   1,                      [DEFAULT]
                                ),

◇ RECH_LINEAIRE               =   _F(
    METHODE                      =   /'CORDE'                      [DEFAULT]
                                /'MIXTE'
                                /'PILOTAGE'
    RESI_LINE_RELA               =   /1.E-1,                      [DEFAULT]
                                /reslin,                      [R]
    ITER_LINE_MAXI               =   /3                      [DEFAULT]
                                /itelin                      [I]
    PAS_MINI_CRIT                =   /0.                      [DEFAULT]
                                /pmicri                      [R]
    ITER_LINE_CRIT               =   /20                      [DEFAULT]
                                /itelic                      [I]
    RHO_MIN                      =   /1.E-2                      [DEFAULT]
                                /rmin                      [R]
    RHO_MAX                      =   /1.E+1                      [DEFAULT]
                                /rmax                      [R]
    RHO_EXCL                     =   /9.E-3                      [DEFAULT]
                                /rexc                      [R]
                                ),

◇ PILOTAGE                    =   _F(
    ◆ TYPE                      =   /'DDL_IMPO',
                                /'SAUT_IMPO',
                                /'LONG_ARC',
                                /'SAUT_LONG_ARC'
                                /'ANA_LIM',
                                /'DEFORMATION',
                                /'PRED_ELAS',
    TOUT                          =   'OUI',                      [DEFAULT]
    /GROUP_MA                    =   lgrma,                      [l_gr_maille]
    /MAILLE                      =   lma,                      [l_maille]
    /NOEUD                      =   no,                      [noeud]

```

```

        /GROUP_NO           =   rno,                      [gr_noeud]
        FISSURE              =   fiss,                    [sd_fiss_xfem],
        NOM_CMP              =   nomcmp,                  [Kn]
        /DIRE_PILO           =   direpilo,                [Kn]
        COEF_MULT            =   /1.,                     [DEFAULT]
                                /cmult,                   [R]
        ETA_PILO_R_MAX       =   etarmax,                 [R]
        ETA_PILO_R_MIN       =   etarmin,                 [R]
        ETA_PILO_MAX         =   etamax,                  [R]
        ETA_PILO_MIN         =   etamin,                   [R]
        PROJ_BORNES          =   /'OUI'                   [DEFAULT]
                                /'NON'
        SELECTION            =   /'NORM_INCR_DEPL',        [DEFAULT]
                                /'ANGL_INCR_DEPL',
                                /'RESIDU',
                                ),
    ◇ SOLVEUR                 =   _F(voir le document [U4.50.01]),
    ◇ CONVERGENCE             =   _F(
        /RESI_GLOB_RELA      =   1.E-6,                  [DEFAULT]
        / | RESI_GLOB_MAXI   =   resmax,                  [R]
        | RESI_GLOB_RELA     =   resrel,                  [R]
        | RESI_COMP_RELA     =   rescmp,                  [R]
        | RESI_REFE_RELA     =   resref,                  [R]
        SIGM_REFE            =   sigref,                  [R]
        EPSI_REFE            =   sigref,                  [R]
        DEPL_REFE            =   depref,                  [R]
        FORC_REFE            =   forref,                  [R]
        VARI_REFE            =   varref,                  [R]
        FLUX_THER_REFE       =   sigref,                  [R]
        FLUX_HYD1_REFE       =   sigref,                  [R]
        FLUX_HYD2_REFE       =   sigref,                  [R]
        ITER_GLOB_ELAS       =   /25,                     [DEFAULT]
                                /maxelas,                [I]
        ITER_GLOB_MAXI       =   /10,                     [DEFAULT]
                                /maglob,                   [I]
        TYPE                 =   /'PIC'                   [DEFAULT]
                                /'PLATEAU'
        PLATEAU_ITER         =   /3                        [DEFAULT]
                                /plaiter                  [I]
        PLATEAU_RELA         =   /1.E-3                    [DEFAULT]
                                /plarel                    [R]
        ARRET                =   /'OUI',                  [DEFAULT]
                                /'NON',
                                ),
    ◇ CRIT_FLAMB              =   _F(
        NB_FREQ              =   /3,                      [DEFAULT]
                                /nbfreq,                  [I]
        CHAR_CRIT            =   /(-10,10),                [DEFAULT]
                                /intcc,
        /LIST_INST           =   list_r8,                  [list_r8]
        /INST                =   l_r8,                     [R]
        /PAS_CALC            =   npas,                     [I]
        ◇ PRECISION          =   /1.E-6,                  [DEFAULT]
                                /prec,                    [R]
        ◇ CRITERE            =   /'RELATIF',               [DEFAULT]
                                /'ABSOLU' , ),
    ◇ SENSIBILITE            =   _F(voir le document [U4.50.02]),
```

```

◇ ARCHIVAGE = _F(
    /LIST_INST = list_r8, [listr8]
    /INST = l_r8, [R]
    /PAS_ARCH = npas, [I]
    ◇ PRECISION = /1.E-6, [DEFAULT]
    /prec, [R]
    ◇ CRITERE = /'RELATIF', [DEFAULT]
    /'ABSOLU' ,
    DETR_NUME_SUIV = 'OUI',
    CHAM_EXCLU = list_txt,
    ),

◇ AFFICHAGE = _F(
    UNITE = /unite [I]
    INFO_RESIDU = /'NON', [DEFAULT]
    /'OUI'
    ),

◇ OBSERVATION = _F( (
    ◆ NOM_CHAM = | 'DEPL',
    | 'VITE',
    | 'ACCE',
    | 'DEPL_ABSOLU',
    | 'VITE_ABSOLU',
    | 'ACCE_ABSOLU',
    | 'SIEF_ELGA',
    | 'VARI_ELGA',
    | 'FORC_NODA',
    | ' VALE_CONT ',
    ◇ EVAL_CHAM = /' VALE ', [DEFAULT]
    /' MIN ',
    /' MAX ',
    /' MOY ',
    ◆ NOM_CMP = lnocmp, [l_Kn]
    ◇ EVAL_CMP = /' VALE ', [DEFAULT]
    /' FORMULE ',

{ Si EVAL_CMP='FORMULE'
    ◇ FORMULE = form [formule_aster]
}
{ Si CHAM est de type ELGA (SIEF_ELGA, VARI_ELGA)
    ◇ TOUT = 'OUI', [DEFAULT]
    ◇ /GROUP_MA = lgrma, [l_gr_maille]
    ◇ / MAILLE = lma, [l_maille]
    ◇ EVAL_ELGA = /' VALE ', [DEFAULT]
    /' MIN ',
    /' MAX ',

    { Si EVAL_ELGA ='VALE'
        ◆ POINT = pi, [I]
        ◇ SOUS_POINT = spi, [I]
    }
}
{ Si CHAM est de type NOEU
    ◇ TOUT = 'OUI', [DEFAULT]
    ◇ /GROUP_MA = lgrma, [l_gr_maille]
    ◇ /MAILLE = lma, [l_maille]
    ◇ /NOEUD = no, [noeud]
    ◇ /GROUP_NO = rno, [gr_noeud]
}
◇ /LIST_INST = linst, [listr8]
◇ /INST = linst, [l_R]
◇ /PAS_OBSE = pas, [I]
```



```

    ◇ CRITERE      =  /'RELATIF',                      [DEFAULT]
                        /'ABSOLU' ,
    { Si CRITERE    =  'RELATIF'
      ◇ PRECISION  =  /1.0E-6,                      [DEFAULT]
                        /prec,                      [R]
    }
    { Si CRITERE    =  'ABSOLU'
      ◇ PRECISION  =  prec,                      [R]
    }
  ),
  ◇ SUIVI_DDL      =  _F (
      ◇ NOM_CHAM   =  |  'DEPL',
                        |  'VITE',
                        |  'ACCE',
                        |  'DEPL_ABSOLU',
                        |  'VITE_ABSOLU',
                        |  'ACCE_ABSOLU',
                        |  'SIEF_ELGA',
                        |  'VARI_ELGA',
                        |  'FORC_NODA',
      ◇ EVAL_CHAM  =  /' VALE ',                      [DEFAULT]
                        /' MIN ',
                        /' MAX ',
                        /' MOY ',
      ◇ NOM_CMP    =  lnocmp ,                      [l_Kn]
      ◇ EVAL_CMP   =  /' VALE ',                      [DEFAULT]
                        /' FORMULE ',

    { Si EVAL_CMP='FORMULE'
      ◇ FORMULE     =  form                      [formule_aster]
    }
    { Si CHAM est de type ELGA (SIEF_ELGA, VARI_ELGA)
      ◇ TOUT        =  'OUI',                      [DEFAULT]
      ◇ /GROUP_MA   =  lgrma,                      [l_gr_maille]
      ◇ /MAILLE     =  lma,                      [l_maille]
      ◇ EVAL_ELGA   =  /' VALE ',                      [DEFAULT]
                        /' MIN ',
                        /' MAX ',
      { Si EVAL_ELGA='VALE'
        ◇ POINT     =  pi ,                      [I]
        ◇ SOUS_POINT =  spi ,                      [I]
      }
    }
    { Si CHAM est de type NOEU
      ◇ TOUT        =  'OUI',                      [DEFAULT]
      ◇ /GROUP_MA   =  lgrma,                      [l_gr_maille]
      ◇ /MAILLE     =  lma,                      [l_maille]
      ◇ /NOEUD      =  no,                      [noeud]
      ◇ /GROUP_NO   =  rno,                      [gr_noeud]
    }
    ◇ TITRE        =  l titre ,                      [ list_k ]
  ),
  ◇ INFO          =  /1,                      [DEFAULT]
                        /2,

  ◇ TITRE          =  tx                      [Kn]
);

```

3 Opérandes

3.1 Opérandes MODELE / CHAM_MATER / CARA_ELEM

♦ `MODELE = mo`

Nom du modèle dont les éléments font l'objet du calcul mécanique.

♦ `CHAM_MATER = chmat`

Nom du champ de matériau affecté sur le modèle `mo`. Attention, toutes les mailles principales du modèle doivent être associées à un matériau (sinon erreur fatale avec message peu explicite),

♦ `CARA_ELEM = carac`

Nom des caractéristiques (`carac`) des éléments de coque, poutre, tuyau, barre, câble, et éléments discrets affectés sur le modèle `mo`. Évidemment, ce mot-clé est optionnel : si le modèle ne contient pas de tels éléments, il n'est pas utile ; en revanche, si le modèle contient de tels éléments, il est obligatoire.

3.2 Mot clé EXCIT

♦ `EXCIT=_F()`

Ce mot clé facteur permet de décrire à chaque occurrence une charge (sollicitations et conditions aux limites), et éventuellement un coefficient multiplicateur et/ou un type de charge.

3.2.1 Opérandes CHARGE

♦ `CHARGE = chi`

`chi` est le chargement mécanique (comportant éventuellement l'évolution d'un champ de température) précisé à la $i^{\text{ème}}$ occurrence de `EXCIT`.

Remarques :

1. Dans un calcul thermo-mécanique, si la température initiale est différente de la température de référence (donnée dans l'opérateur `AFFE_MATERIAU`), le champ de déformation associé à l'instant initial peut être incompatible et donc conduire à un état de contraintes et de variables internes associé non nul. Si l'on utilise une relation de comportement incrémentale (mot clé facteur `COMP_INCR`) et si on ne définit pas explicitement un état de contraintes et de variables internes initial (associé à un champ de température initiale différente de la température de référence), le champ de contraintes et de variables internes calculé au premier incrément ne tiendra compte que de la seule variation de température entre l'instant initial et le premier instant, et non des éventuelles contraintes de compatibilité associées à la température initiale. Pour prendre cet état initial en compte, il faut le donner explicitement, par exemple grâce aux mots clés `SIGM`, `DEPL`, `VARI` et `VARI_NON_LOCAL` dans `ETAT_INIT`. **Pour éviter de telles situations qui peuvent conduire à des erreurs de calculs, il vaut mieux commencer un calcul en considérant qu'il faut partir d'un état vierge.**

2. Si on réalise un calcul en axisymétrie et que l'on impose des forces nodales, ces efforts doivent être divisés par 2π (on travaille sur un secteur d'un radian) par rapport aux chargements réels. De même, si l'on souhaite calculer la résultante des efforts, le résultat est à multiplier par 2π pour avoir la résultante totale sur la structure complète. De même en contraintes planes ou en déformation plane, on travaille sur une épaisseur unité : les efforts (sur l'épaisseur) appliqués doivent être divisés par l'épaisseur, les efforts réels sont obtenus en multipliant par l'épaisseur les efforts « du calcul ».

3.2.2 Opérande FONC_MULT

◇ $\text{FONC_MULT} = f_i$

f_i est la fonction du temps multiplicatrice du chargement précisé à la $i^{\text{ème}}$ occurrence de EXCIT.

Le chargement et les conditions aux limites pour n occurrences du mot clé facteur EXCIT sont :

$$ch = \sum_{i=1}^n f_i \cdot ch_i$$

Pour les conditions de Dirichlet, bien entendu, seule la valeur imposée est multipliée par f_i .

Par défaut : $f_i = 1$.

3.2.3 Opérande TYPE_CHARGE

◇ $\text{TYPE_CHARGE} = \text{'FIXE_CSTE'}$, [DEFAULT]
 /'SUIV',
 /'DIDI',
 /'FIXE_PILO'

Par défaut, tchi vaut 'FIXE_CSTE' : cela correspond à un chargement appliqué sur la géométrie initiale et non piloté. Il peut cependant être une fonction, et, en particulier, dépendre du temps.

Si tchi vaut 'FIXE_PILO', le chargement est toujours fixe (indépendant de la géométrie) mais sera piloté grâce au mot clé PILOTAGE [§20]. Les charges pilotables doivent être issues d'AFFE_CHAR_MECA ou d'AFFE_CHAR_MECA_F (si ce n'est pas une fonction dépendant du temps) et ne pas être affectées du mot clé FONC_MULT. On ne peut pas piloter les chargements de pesanteur, la force centrifuge, les forces de Laplace, les chargements thermiques ou de déformations initiales ou anélastiques, et les conditions de liaison.

Si tchi vaut 'SUIV', le chargement est dit « suiveur », c'est-à-dire qu'il dépend de la valeur des inconnues : par exemple, la pression, étant un chargement s'appliquant dans la direction normale à une structure, dépend de la géométrie actualisée de celle-ci, et donc des déplacements. Un chargement suiveur est réévalué à chaque itération de l'algorithme de résolution. Un chargement fixe n'est réévalué qu'à chaque nouvel instant, et seulement si ch_i dépend du temps (défini dans AFFE_CHAR_MECA_F et paramétré par l'instant).

Actuellement les chargements qui peuvent être qualifiés de 'SUIV' sont le chargement de pesanteur pour l'élément de CABLE_POULIE, la pression pour les modélisations 3D, 3D_SI, D_PLAN, D_PLAN_SI, AXIS, AXIS_SI, C_PLAN, C_PLAN_SI et pour toutes les modélisations THM (3D_HHM, 3D_HM, 3D_JOINT_CT, 3D_THH, 3D_THHM, 3D_THM, AXIS_HHM, AXIS_HM, AXIS_THH, AXIS_THHM, AXIS_THM, D_PLAN_HHM, D_PLAN_HM, D_PLAN_THH, D_PLAN_THHM, D_PLAN_THM) et la force centrifuge en grands déplacements (mot clé ROTATION dans AFFE_CHAR_MECA).

Si tchi vaut 'DIDI' alors les conditions de Dirichlet (déplacements imposés, conditions linéaires) s'appliqueront sur l'incrément de déplacement à partir de l'instant donné sous ETAT_INIT/NUME_DIDI (par défaut l'instant de reprise du calcul) et non sur le déplacement total. Par exemple pour un déplacement imposé (mot clé DDL_IMPO de AFFE_CHAR_MECA) la condition sera de la forme $u - u_0 = d$ où u_0 est le déplacement défini par NUME_DIDI et non $u = d$.

3.3 Mot clé CONTACT

◇ $\text{CONTACT} = \text{contact}$

Ce mot clé simple permet d'activer la résolution du contact-frottement ou la prise en compte d'une liaison unilatérale. `contact` est un concept issu de l'opérateur `DEFI_CONTACT` [U4.44.11].

Attention :

*Ce mot-clé simple n'accepte qu'un seul concept. On ne peut donc pas mélanger dans un même calcul non-linéaire la résolution du contact et la prise en compte d'une liaison unilatérale. On ne peut pas non plus mélanger les différentes formulations (*DISCRETE*, *CONTINUE* et *XFEM*)*

3.4 Mot-clé SOUS_STRUC

◇ `SOUS_STRUC`

Ce mot clé facteur permet de préciser quels sont les chargements à utiliser pour les sous-structures statiques qui font alors obligatoirement partie du modèle. En son absence, les chargements sur les sous structures sont nuls.

Ces chargements s'ajoutent aux chargements « éléments finis » qui peuvent être appliqués sur le reste du modèle. Pour plus de précision concernant l'utilisation de sous-structures (élastiques linéaires) dans une structure non-linéaire, on se reportera à la documentation [U2.07.02] et le cas-test `ssnv193a`.

3.4.1 Opérande CAS_CHARGE

◆ `CAS_CHARGE = nocas`

`nocas` est le nom du cas de charge à utiliser. Voir opérateur `MACR_ELEM_STAT` [U4.62.01].

3.4.2 Opérandes TOUT / SUPER_MAILLE

◆ `/TOUT = 'OUI'`

Ce mot clé permet d'affecter le chargement `nocas` à toutes les sous structures du modèle.

`/SUPER_MAILLE = l_mail`

Ce mot clé facteur permet de n'affecter le chargement `nocas` qu'à certaines sous-structures.

3.4.3 Opérande FONC_MULT

◇ `FONC_MULT = f_i`

f_i est la fonction du temps multiplicatrice du chargement précisé à la $i^{\text{ème}}$ occurrence de `SOUS_STRUCT`.

Le comportement de ce mot clé est le même que pour son occurrence dans `EXCIT`.

3.5 Mots-clés COMP_INCR et COMP_ELAS

La syntaxe de ces mots-clés communs à plusieurs commandes est décrite dans le document [U4.51.11].

3.6 Mot clé ETAT_INIT

◇ `ETAT_INIT`

Ce mot-clé permet de définir un état initial de référence. Par défaut, tous les champs sont identiquement nuls. L'état initial peut être défini soit en précisant chaque champ de l'état initial, soit en extraction depuis un concept de type `evol_noli` préexistant, soit en mélangeant les deux. La donnée d'un état initial n'a de sens (et n'est donc prise en compte) que pour la partie du domaine traitée en comportement incrémental (`COMP_INCR`) ; si le comportement est élastique (`COMP_ELAS`) cela n'a aucune incidence.

Si on utilise simultanément `EVOL_NOLI` et un (des) champ(s) individuels par `SIGM/VARI/DEPL`, le principe de surcharge s'applique selon la règle suivante : on prend tous les champs disponibles dans `evol`, puis on surcharge avec les champs données individuellement

Si l'on veut prendre en compte un état initial en élasticité, c'est le mot clé `ELAS` situé sous `COMP_INCR` qu'il faut utiliser.

Remarques :

- Dans le cas où l'utilisateur a spécifié que le concept résultat est réentrant (par le mot réservé `reuse`), le mot-clé `ETAT_INIT` est obligatoire.
- Dans le cas où l'on utilise la méthode continue du contact, la reprise de calcul peut donner lieu à des difficultés de convergence du fait de « l'oubli » de l'état de contact précédent.

3.6.1 Opérandes `SIGM/VARI/DEPL/STRX`

```
♦ / | SIGM = sig
    | VARI = vain
    | DEPL = depl
    | STRX = strx
```

`sig` est le champ de contraintes aux points de Gauss, `vain` est le champ des variables internes aux points de Gauss, `depl` est le champ des déplacements aux nœuds pris à l'état initial et `strx` est le champ d'efforts et de déplacements correspondant aux éléments de structures. Si l'un de ces champs n'est pas précisé, il est pris nul par défaut. Ils peuvent par exemple être issus de la commande `CREA_CHAMP`, ou bien avoir été lus dans un fichier au format I-DEAS ou MED par la commande `LIRE_RESU`.

3.6.2 Opérandes `EVOL_NOLI`

```
/ EVOL_NOLI = evol
```

Nom du concept de type `evol_noli` d'où sera extrait l'état initial.

3.6.3 Opérande `NUME_ORDRE / INST / NUME_DIDI`

```
♦ / NUME_ORDRE = nuini
  / INST       = instini
```

Extraction de l'état mécanique initial dans `evol` à partir du numéro d'archivage `NUME_ORDRE` ou de l'instant d'archivage `INST` pour effectuer la poursuite du calcul.

Si `NUME_ORDRE` ou `INST` ne sont pas remplis, on prend le dernier numéro archivé existant dans `evol`.

```
♦ NUME_DIDI = nudidi
```

Dans le cas de chargements de type Dirichlet différentiel ('`DIDI`'), on donne sous `NUME_DIDI` le numéro d'archivage de l'état mécanique (déplacements) qui sert de référence pour l'application de ces conditions aux limites. Par défaut on prend l'état mécanique défini sous `NUME_ORDRE` ou `INST`.

3.6.4 Opérande `INST_ETAT_INIT`

```
♦ INST_ETAT_INIT = istetaini
```

On peut associer une valeur d'instant `istetaini` à cet état initial.

Par défaut :

- 1) Lorsque l'état initial est défini par la donnée des champs (`ETAT_INIT` avec `DEPL/SIGM/VARI`), il n'y a pas d'instant associé.
- 2) Lorsque l'état est donné par un concept `evol_noli` (`ETAT_INIT` avec `EVOL_NOLI`), il s'agit de l'instant dans le précédent calcul (`istetaini = instini`).

A - Exemple simple (comportement par défaut)

```
LIST1 = DEFI_LIST_REEL(DEBUT =0.,  
                        INTERVALLE =_F(JUSQU'A = 4., NOMBRE =4)),  
  
U      = STAT_NON_LINE (INCREMENT =_F(LIST_INST =LIST1)) ,  
  
LIST2 = DEFI_LIST_REEL(DEBUT =4.,  
                        INTERVALLE =_F(JUSQU'A = 10., NOMBRE =6)),  
  
U      = STAT_NON_LINE (reuse=U,  
                        INCREMENT =_F(LIST_INST =LIST2),  
                        ETAT_INIT =_F(EVOL_NOLI =U)) ,
```

Premier STAT_NON_LINE : effectue le calcul pour les instants 1, 2, 3 et 4s.

Second STAT_NON_LINE : effectue le calcul pour les instants 5, 6, 7, 8, 9 et 10s, l'état initial correspondant au temps 4s.

B - Exemple pour montrer l'intérêt de INST_ETAT_INIT (deux listes d'instantes différentes)

```
LIST1 = DEFI_LIST_REEL(DEBUT =0.,  
                        INTERVALLE =_F(JUSQU'A = 10., NOMBRE =10)),  
  
U      = STAT_NON_LINE (INCREMENT =_F(LIST_INST =LIST1)) ,  
  
LIST2 = DEFI_LIST_REEL(DEBUT =20.,  
                        INTERVALLE =_F(JUSQU'A = 30., NOMBRE =10)),  
  
U      = STAT_NON_LINE (reuse=U  
                        INCREMENT =_F(LIST_INST =LIST2),  
                        ETAT_INIT =_F(EVOL_NOLI =U,  
                        INST_ETAT_INIT = 20.)) ,
```

Premier STAT_NON_LINE : effectue le calcul des instants 1 à 10s.

Second STAT_NON_LINE : effectue le calcul des instants 21 à 30s, l'état initial correspondant à l'instant $t=10s$ du premier STAT_NON_LINE (par défaut INST=10.). Cet état initial correspond pour ce second STAT_NON_LINE à l'instant $t=20s$. (INST_ETAT_INIT=20.).

C - Exemple pour montrer l'intérêt de INST_ETAT_INIT (pratique quand on fait du cyclique)

```
LIST1 = DEFI_LIST_REEL(DEBUT =0.,  
                        INTERVALLE =_F(JUSQU'A = 10., NOMBRE =10)),  
  
U1     = STAT_NON_LINE (INCREMENT =_F(LIST_INST =LIST1)) ,  
  
U2     = STAT_NON_LINE (INCREMENT =_F(LIST_INST =LIST1),  
                        ETAT_INIT =_F(EVOL_NOLI =U1,  
                        INST_ETAT_INIT = 0.)) ,
```

Premier STAT_NON_LINE : effectue le calcul des instants 1 à 10s.

Second STAT_NON_LINE : effectue le calcul des instants 1 à 10s, l'état initial correspondant à l'instant $t=10s$ du premier STAT_NON_LINE (par défaut INST=10.). Cet état initial correspond pour ce second STAT_NON_LINE à l'instant $t=0s$. (INST_ETAT_INIT= 0.).

3.6.5 Opérande PRECISION / CRITERE

◇ PRECISION = prec

Cf. [U4.71.00] pour la syntaxe détaillée

Ce paramètre sert à repérer le bon numéro d'ordre (NUME_ORDRE) quand l'utilisateur renseigne l'instant (INST). En effet, les instants dans STAT_NON_LINE sont repérés par un numéro d'ordre (un

entier). Si l'utilisateur veut utiliser un instant (un réel) et non un numéro d'ordre pour `INST`, l'opérande précision permet de sélectionner ce numéro d'ordre.
Exemple:

NUME	1	2	3	4	5	6	7
INST	0.0010	0.0020	0.0030	0.0040	0.0050	0.0060	0.0070

Si l'utilisateur veut sélectionner l'instant correspondant à `NUME=4`, il lui suffit de dire `INST=0,004`. Par contre, pour le deuxième exemple:

NUME	1	2	3	4	5	6	7
INST	0.10000001	0.10000002	0.10000003	0.10000004	0.10000005	0.10000006	0.10000007

Si l'utilisateur veut sélectionner l'instant correspondant à `NUME=4`, il ne lui suffit pas de dire `INST=0,10000004`, car l'écart relatif entre les instants vaut $\frac{0,10000005 - 0,10000004}{0,10000004} = 1E-7$

qui est supérieur à la valeur de précision par défaut ($1E-6$). On ne pourra donc pas distinguer `NUME=3,4` et `5` (le code s'arrête alors en erreur fatale). Il suffit alors de changer le paramètre `PRECISION` pour pouvoir sélectionner l'instant (dans l'exemple, `PRECISION=1E-8` conviendra).

3.7 Mot clé INCREMENT

◆ `INCREMENT=_F()`

Définit les intervalles de temps pris dans la méthode incrémentale.

Les instants ainsi définis n'ont de sens physique que pour des relations de comportement où le temps intervient explicitement (visco-élastiques ou visco-plastiques par exemple). Dans les autres cas, ils permettent seulement d'indiquer les incréments de charge et de paramétrer l'évolution d'un éventuel champ de température.

3.7.1 Opérande LIST_INST

◆ `LIST_INST = /litpsr8, [listr8]
 /litps, [list_inst]`

◆ Si `LIST_INST = litpsr8 [listr8]`

Les instants de calcul sont ceux définis dans le concept `litpsr8` par l'opérateur `DEFI_LIST_REEL` [U4.34.01].

◆ Si `LIST_INST = litps [list_inst]`

Les instants de calcul sont ceux définis dans le concept `litps` par l'opérateur `DEFI_LIST_INST` [U4.34.03].

3.7.2 Opérandes NUME_INST_INIT / INST_INIT / NUME_INST_FIN / INST_FIN

`/NUME_INST_INIT = nuini
/INST_INIT = instini`

L'instant initial du calcul (qui donc n'est pas (re)calculé) est désigné soit par sa valeur (`INST_INIT`), soit par son numéro d'ordre dans la liste d'instants `litps` (`NUME_INST_INIT`). Pour pouvoir accéder par valeur, il est nécessaire que la liste soit ordonnée.

En l'absence des mots clés `INST_INIT` ou `NUME_INST_INIT`, le défaut est calculé de la manière suivante :

- 1) Si un état initial est précisé (opérande ETAT_INIT) et s'il définit un instant correspondant (par EVOL_NOLI ou INST_ETAT_INIT) alors l'instant initial est celui défini par cet état initial,
- 2) S'il n'y a pas d'état initial (opérande ETAT_INIT absent) ou qu'il ne définit pas d'instant correspondant (les champs sont donnés dans ETAT_INIT sans préciser INST_ETAT_INIT), alors on prend le premier instant de la liste d'instants (NUME_INST_INIT=0).
- 3) En cas d'archivage (voir mot-clef ARCHIVAGE), l'instant initial en poursuite est le dernier pas archivé et non celui défini dans INST_INIT.

```
/NUME_INST_FIN = nufin  
/INST_FIN      = instfin
```

L'instant final (dernier pas calculé) est désigné de la même manière que l'instant initial (soit NUME_INST_FIN, soit INST_FIN), sauf qu'il n'est pas possible de faire référence à l'instant de l'état initial.

Attention :

- Si le re-découpage automatique du pas de temps est activé, NUME_INST_FIN n'en tient pas compte et travaille toujours sur la liste d'instants initial. NUME_INST_INIT et NUME_INST_FIN ne sont actifs qu'à l'initialisation.

A - Exemple simple (comportement par défaut)

```
LIST = DEFI_LIST_REEL( DEBUT =0.,  
                       INTERVALLE =_F(JUSQU'A= 10., NOMBRE =10)),  
  
U = STAT_NON_LINE (    INCREMENT =_F ( LIST_INST =LIST,  
                                       INST_FIN =4.)) ,  
  
U = STAT_NON_LINE(    reuse=U,  
                     INCREMENT =_F ( LIST_INST =LIST),  
                               ETAT_INIT =_F (EVOL_NOLI :U)) ,
```

Premier STAT_NON_LINE : effectue le calcul pour les instants 1, 2, 3 et 4s.

Second STAT_NON_LINE : effectue le calcul pour les instants 5, 6, 7, 8, 9 et 10s, l'état initial correspondant au temps 4s. (par défaut INST_INIT=INST_ETAT_INIT=INST=4.).

B - Exemple pour montrer l'intérêt de INST_INIT

```
LIST = DEFI_LIST_REEL( DEBUT =0.,  
                       INTERVALLE =_F (JUSQU'A = 10., NOMBRE =10)),  
  
U = STAT_NON_LINE (    INCREMENT =_F (LIST_INST = LIST,  
                                       INST_FIN = 4.)) ,  
  
U = STAT_NON_LINE (    reuse = U,  
                     INCREMENT =_F ( LIST_INST =LIST,  
                               INST_INIT =8.),  
                               ETAT_INIT =_F ( EVOL_NOLI =U)) ,
```

Premier STAT_NON_LINE : effectue le calcul des instants 1 à 4s.

Second STAT_NON_LINE : effectue le calcul pour les instants 9 et 10s (ne fait rien pour $t=5,6,7$ et 8s), l'état initial correspondant au temps $t=4s$ (par défaut INST=4.).

3.7.3 Opérande PRECISION

◇ PRECISION = prec

Cf. [U4.71.00] pour la syntaxe détaillée

Ce paramètre sert à repérer le bon numéro d'ordre (NUME_INST_FIN/NUME_INST_INIT) quand l'utilisateur renseigne l'instant (INST_FIN/INST_INIT). En effet, les instants dans STAT_NON_LINE

sont repérés par un numéro d'ordre (un entier). Si l'utilisateur veut utiliser un instant (un réel) et non un numéro d'ordre pour (NUME_INST_*), l'opérande précision permet de sélectionner ce numéro d'ordre.
Exemple:

NUME	1	2	3	4	5	6	7
INST	0.0010	0.0020	0.0030	0.0040	0.0050	0.0060	0.0070

Si l'utilisateur veut sélectionner l'instant correspondant à NUME=4, il lui suffit de dire INST=0,10000004. Par contre, pour le deuxième exemple:

NUME	1	2	3	4	5	6	7
INST	0.10000001	0.10000002	0.10000003	0.10000004	0.10000005	0.10000006	0.10000007

Si l'utilisateur veut sélectionner l'instant correspondant à NUME=4, il ne lui suffit pas de dire INST=0,10000004, car l'écart relatif entre les instants vaut $\frac{0,10000005 - 0,10000004}{0,10000004} = 1E-7$

qui est supérieur à la valeur de précision par défaut (1E-6). On ne pourra donc pas distinguer NUME=3, 4 et 5 (le code s'arrête alors en erreur fatale). Il suffit alors de changer le paramètre PRECISION pour pouvoir sélectionner l'instant (dans l'exemple, PRECISION=1E-8 conviendra).

3.7.4 Opérande ERRE_TEMPS

◇ ERRE_TEMPS = / 'NON' [DEFAULT]
/ 'OUI'

Cet opérande permet d'activer le calcul des indicateurs d'erreur temporel pour les modélisations HM stationnaires. Voir [R4.10.05].

3.8 Opérande METHODE

◇ METHODE = / 'NEWTON'
/ 'IMPLEX'

Permet de choisir la méthode de résolution du problème incrémental non linéaire.

/ 'NEWTON'

On utilise l'algorithme de Newton-Raphson pour résoudre le problème (voir [R5.03.01]).

/ 'IMPLEX'

On utilise l'algorithme IMPLEX pour résoudre le problème (voir [R5.03.81]).

3.9 Mot clé NEWTON

Précise les caractéristiques de la méthode de résolution du problème incrémental non-linéaire (méthode de Newton-Raphson)

3.9.1 Opérande PREDICTION

◇ PREDICTION = / 'TANGENTE'
/ 'ELASTIQUE'
/ 'EXTRAPOL'
/ 'DEPL_CALCULE'

La phase de prédiction (Cf. [R5.03.01]) a pour but de calculer une estimation du champ de déplacements afin de permettre à la méthode de Newton de converger plus rapidement.

Lorsque le mot clé est absent, c'est la matrice tangente en vitesse (option `RIGI_MECA_TANG`) qui est utilisée si l'on a choisi pour la méthode de Newton une `MATRICE='TANGENTE'`, et c'est la matrice élastique (option `RIGI_MECA`) qui est utilisée si on a choisi `MATRICE='ELASTIQUE'`.

`/ 'TANGENTE'`

On utilise la matrice tangente du problème en vitesse (option `RIGI_MECA_TANG`).

`/ 'ELASTIQUE'`

On utilise la matrice élastique (option `RIGI_MECA`).

`/ 'EXTRAPOL'`

On calcule l'estimation de l'incrément de déplacement à partir de l'incrément total obtenu comme solution au pas de temps précédent (pondéré par le rapport des pas de temps). On projette cette estimation sur l'ensemble des champs cinématiquement admissibles (i.e. satisfaisant les conditions aux limites de Dirichlet) selon la norme donnée par la matrice élastique, qui doit donc être calculée. Cette fonctionnalité est intéressante dans le cas de l'utilisation de schémas d'intégration locale explicite de type Runge-Kutta qui ne fournissent pas de matrice tangente : dans ce cas la méthode de Newton utilise une matrice élastique, mais le nombre d'itérations nécessaires peut être élevé. L'utilisation de l'extrapolation peut améliorer les performances.

`/ 'DEPL_CALCULE'`

Permet de proposer comme déplacement pour la prédiction à chaque pas de temps, le déplacement donné par une histoire mécanique précisée sous le mot clé `EVOL_NOLI` (§13). Le déplacement est projeté sur l'ensemble des champs cinématiquement admissible, comme pour la méthode `EXTRAPOLE`.

Remarques:

- Les méthodes `'EXTRAPOLE'` et `'DEPL_CALCULE'` procèdent à une projection de la solution sur l'ensemble des champs cinématiquement admissibles. On se sert pour cela des conditions aux limites de Dirichlet donné dans le mot-clé `EXCIT`. Dans ce cas, il n'est pas possible d'utiliser des chargements de Dirichlet de type « cinématique » (opérande `AFFE_CHAR_CINE`) mais uniquement des chargements de Dirichlet par dualisation (opérande `AFFE_CHAR_MECA`). Une alarme prévient l'utilisateur dans le cas où `Code_Aster` n'aurait pas trouvé de chargements de Dirichlet dualisés. Le risque dans ce cas étant que le champ de déplacement ne soit pas cinématiquement admissible.
- Du fait de cette projection sur les conditions limites, ces deux options sont incompatibles avec les fonctionnalités du `PILOTAGE` ;

Utilité :

- 1) supposons qu'on réalise un premier calcul avec un maillage grossier. On souhaite réaliser le même calcul mais sur un maillage plus fin. On peut supposer que la solution en déplacement pour ce second calcul n'est pas éloignée de celle du premier calcul et donc qu'une bonne prédiction du déplacement pour ce second calcul est la projection des déplacements du premier calcul sur les nœuds du nouveau maillage (la projection des déplacements sur le nouveau maillage doit être réalisée préalablement avec l'opérateur `PROJ_CHAMP` [U4.72.05]). Ce mot clé permet de réaliser ce mode de prédiction.
- 2) cela permet de réduire la place mémoire et de conserver ces résultats en vue d'une poursuite ultérieure. Pour un gros calcul, on peut stocker uniquement les déplacements à tous les instants aux formats `IDEAS` ou `MED` dans `IMPR_RESU`. Si on veut recalculer les contraintes et variables internes, on fait un `LIRE_RESU` au format adéquat puis on utilise `DEPL_CALCULE` avec `ITER_GLOB_MAXI=0` (on effectue une seule itération) et `ARRET='NON'` (il n'y a pas convergence, on ne vérifie pas l'équilibre). Il est toutefois nécessaire pour des raisons de syntaxe de donner un chargement (éviter les chargements Dirichlet qui imposent une résolution linéaire) ainsi qu'un critère de convergence, même si ces informations ne sont pas prises en compte.

3.9.2 Opérande MATRICE

```
◇ MATRICE = / 'TANGENTE'  
◇ REAC_INCR = /1 [DEFAULT]  
/mf  
◇ REAC_ITER = /0 [DEFAULT]
```

/it

La matrice utilisée pour les itérations globales de la méthode est la matrice tangente [R5.03.01]. La matrice tangente de prédiction est réévaluée tous les `mf` incréments de temps (`mf` positif ou nul) et la matrice tangente cohérente (option `FULL_MECA`) est réévaluée toutes les `it` itérations de Newton pour un incrément de temps donné (précisément aux itérations de numéro `it`, `2it`, `3it`...). Donc à la première itération de Newton, on ne réassemble la matrice tangente que si `it` vaut 1 : sinon on garde la matrice utilisée dans la phase de prédiction. Par convention si `it` vaut 0 la matrice n'est pas réévaluée durant tout le pas de temps.

```
◇ PAS_MINI_ELAS = /0. [DEFAULT]
                  /pasmini [R]
◇ REAC_ITER_ELAS = /0 [DEFAULT]
                  /it [I]
```

Ces options permettent de passer de la matrice tangente à la matrice de décharge (i.e en considérant que les non linéarités n'évoluent pas) lorsque le pas de temps est inférieur à `pasmini`. Cette matrice de décharge est la matrice élastique pour les modèles de comportement de type plastique ; pour les modèles d'endommagement elle s'identifie à la matrice sécante.

Comme la convergence avec la matrice élastique est plus lente que celle avec la matrice tangente, le mot clé `ITER_GLOB_ELAS` sous le mot clé facteur `CONVERGENCE` permet de définir un nombre d'itérations maximal spécifique à l'utilisation de la matrice élastique et différent de celui associé à l'utilisation de la matrice tangente.

On peut définir une fréquence de réactualisation de la matrice de décharge avec le mot-clé `REAC_ITER_ELAS` (analogue de `REAC_ITER`). Si la matrice de décharge ne dépend pas de l'état de déformation (ce qui est le cas pour les matériaux plastiques mais pas pour les modèles d'endommagement), prendre `REAC_ITER_ELAS = 0` (puisque'elle sera la même au cours des itérations).

Utilité :

Cette option peut être utile lorsque le redécoupage automatique du pas de temps ne suffit pas à faire converger un calcul. Par exemple, dans le cas de lois adoucissantes, la matrice tangente peut devenir singulière et il vaut donc mieux utiliser la matrice élastique pour converger.

```
◇ MATRICE = /'ELASTIQUE'
```

La matrice utilisée correspond au calcul élastique : elle n'est évaluée qu'une fois à l'instant initial, en début d'algorithme. Cette matrice "élastique" est calculée en utilisant le module d'Young donné sous le mot clé `ELAS` de l'opérateur `DEFI_MATERIAU`, et non pas la pente à l'origine de la courbe de traction donnée sous le mot clé `TRACTION` (et qui sert, elle, dans l'expression des relations de comportement `VMIS_ISOT_TRAC`, `ECMI_ISOT_TRAC`, `VISC_ISOT_TRAC` [U4.51.11]).

3.9.3 Opérande EVOL_NOLI

```
◇ EVOL_NOLI = evol_noli
```

Nom du concept de type `evol_noli` qui servira dans la prédiction par `DEPL_CALCULE`.

3.10 Mot clé RECH_LINEAIRE

```
◇ RECH_LINEAIRE=_F()
```

La recherche linéaire peut permettre d'améliorer la convergence de la méthode de Newton (Cf. [R5.03.01] pour plus de détails).

Attention :

Il est déconseillé d'utiliser la recherche linéaire avec les déformations `GROT_GDEP` pour les modélisations `COQUE_3D` et en présence de contact.

3.10.1 Opérande METHODE

```
◇ METHODE = /'CORDE' [DEFAULT]
            /'MIXTE'
            /'PILOTAGE'
```

Permet de choisir la méthode de recherche linéaire, c'est-à-dire l'algorithme de recherche du zéro de la fonctionnelle (voir doc [R5.03.01]). La méthode CORDE (par défaut) est la méthode la plus simple, c'est une méthode sécante unidimensionnelle.

La méthode MIXTE est plus élaborée et utilise une méthode sécante avec des bornes variables. Elle est plus efficace lorsque la fonctionnelle n'est pas strictement concave (problèmes avec endommagement ou THM par exemple).

La méthode PILOTAGE est réservée au pilotage de type DEFORMATION, PRED_ELAS et LONG_ARC (voir §20). C'est la seule méthode utilisable avec ce type de pilotage. Pour le pilotage de type DDL_IMPO, on peut utiliser CORDE ou MIXTE.

3.10.2 Opérande RESI_LINE_RELA / ITER_LINE_MAXI

```
◇ RESI_LINE_RELA = /1.E-1          [DEFAULT]
                  /reslin
◇ ITER_LINE_MAXI = /3              [DEFAULT]
                  /itelin
```

Ce sont les paramètres de la recherche linéaire. On donne le nombre d'itérations maximum *itelin* à effectuer et la précision *reslin* à atteindre pour réaliser la convergence de la recherche linéaire. Il est conseillé de ne pas utiliser la recherche linéaire avec du contact.

Pour la méthode CORDE, il n'est pas nécessaire de spécifier une précision ni un nombre d'itérations très élevés, la pratique montrant que deux ou trois itérations de recherche linéaire sont suffisantes. On peut donc se contenter de demander trois itérations avec la précision par défaut. Vous ne pouvez mettre plus de 999 itérations de recherche linéaire pour la méthode CORDE.

Par contre, pour la méthode MIXTE, sur des problèmes avec endommagement, plusieurs dizaines d'itérations sont souvent efficaces.

3.10.3 Opérande PAS_MINI_CRIT / ITER_LINE_CRIT

```
◇ PAS_MINI_CRIT   = /0.            [DEFAULT]
                  /pmicri          [R]
◇ ITER_LINE_CRIT  = /20            [DEFAULT]
                  /itelic          [I]
```

Lors de pas de temps où la convergence est délicate, on peut vouloir augmenter le nombre maximum d'itérations de recherche linéaire. C'est ce que permettent les mots-clés PAS_MINI_CRIT et ITER_LINE_CRIT. Quand le pas de temps (directement fixé par l'utilisateur ou conséquence de découpages de pas de temps) devient inférieur à la valeur *pmicri*, le nombre d'itérations de recherche de recherche linéaire passe de *itelin* (renseigné par ITER_LINE_MAXI) à *itelic* (renseigné par ITER_LINE_CRIT).

3.10.4 Opérandes RHO_MIN/RHO_MAX/RHO_EXCL

```
◇ RHO_MIN         = / 1.E-2        [DEFAULT]
                  / rmin           [R]
◇ RHO_MAX         = / 1.E+1        [DEFAULT]
                  / rmax           [R]
◇ RHO_EXCL        = / 9.E-3        [DEFAULT]
                  / rexc           [R]
```

Ces mots-clés fixent l'intervalle *I* dans lequel on calcule le coefficient RHO de la recherche linéaire, sous la forme : $I = [rmin, rmax] - [-rexc, rexc]$ [R5.03.01].

3.11 Mot clé PILOTAGE

```
◇ PILOTAGE = _F()
```

Lorsque l'intensité η d'une partie du chargement n'est pas connue a priori (chargement dit de référence défini dans AFFE_CHAR_MECA ou AFFE_CHAR_MECA_F avec charge de type FIXE_PILLO), le mot clé PILOTAGE permet de piloter ce chargement par l'intermédiaire d'un nœud (ou groupe de nœud) sur lequel on peut imposer différents modes de pilotage (mot clé TYPE).

Attention :

- Avec *FIXE_PILLO*, on ne peut pas utiliser pour le chargement de référence le mot clé *FONC_MULT*.
- Lorsque le chargement de référence est défini par *AFFE_CHAR_MECA_F*, ce chargement peut être fonction des variables d'espace mais pas du temps. De même, les changements issus de variables de commande (comme la température) qui dépendent du temps ne sont pas utilisables avec le pilotage.
- Le mot clé *PILOTAGE* est interdit avec le contact (sauf dans le cas du contact *XFEM*).
- Il n'est pas possible de faire du *PILOTAGE* avec *PREDICTION='DEPL_CALCULE'* ou *PREDICTION='EXTRAPOLE'* (voir § 17)

3.11.1 Opérande TYPE

```
◇ TYPE = / 'DDL_IMPO'  
          / 'LONG_ARC '  
          / 'ANA_LIM'  
          / 'DEFORMATION'  
          / 'PRED_ELAS '  
          / 'SAUT_IMPO '  
          / 'SAUT_LONG_ARC '
```

C'est le type de pilotage effectué. Sept modes de pilotage sont disponibles (Cf. [R5.03.80] pour plus de détails) :

/ 'DDL_IMPO'

Permet d'imposer une valeur donnée d'incrément de déplacement (une seule composante i possible) en un unique nœud no (ou d'un groupe de nœuds ne comportant qu'un seul nœud). À chaque incrément de temps, on cherche l'amplitude η du chargement de référence qui permettra de satisfaire la relation incrémentale suivante :

$$cmult . \Delta u_i(no) = \Delta t$$

/ 'SAUT_IMPO'

Reprend le principe de *DDL_IMPO* mais pour contrôler l'incrément du saut de déplacement entre les lèvres d'une fissure X-FEM. Une seule direction i est possible, mais elle peut être définie dans une base locale (normale ou tangente à la fissure). On contrôle la moyenne de cet incrément de saut sur un ensemble de points d'intersection P_a de l'interface avec les arêtes a du maillage. Cet ensemble décrit toute la fissure si *GROUP_NO* n'est pas renseigné (comportement par défaut), et seulement une partie s'il l'est. Attention, ce type de pilotage ne peut être utilisé qu'en modélisation X-FEM.

$$cmult . \frac{1}{N} \sum_{a=1}^N \llbracket \Delta u_i \rrbracket (P_a) = \Delta t$$

/ 'LONG_ARC'

Permet de piloter l'intensité η du chargement de référence par la longueur (abscisse curviligne) de la réponse en déplacement d'un groupe de nœuds (à utiliser par exemple lorsqu'on veut contrôler le flambement d'une éprouvette). On vérifie la relation suivante :

$$cmult . \|\Delta u\| = \Delta t \text{ avec } \|(\Delta u)\| = \left(\sqrt{\sum_n \sum_c (\Delta u_{n,c}^2)} \right)$$

où n sont les nœuds du pilotage et c les composantes du déplacement des nœuds considérés. Même si le groupe de nœud du pilotage est réduit à un seul nœud, il faut quand même utiliser *GROUP_NO*.

/'SAUT_LONG_ARC'

Reprend le principe de LONG_ARC mais pour contrôler la norme de l'incrément du saut de déplacement entre les lèvres d'une fissure X-FEM. On contrôle cette norme en moyenne sur un ensemble de points d'intersection P_a de l'interface avec les arêtes a du maillage. Cet ensemble décrit toute la fissure si GROUP_NO n'est pas renseigné (comportement par défaut), et seulement une partie s'il l'est.

$$\text{cmult} . |||\bar{\Delta u}||| = \Delta t \text{ avec } |||\bar{\Delta u}||| = \sqrt{\frac{1}{N} \sum_c \sum_{a=1}^N (||\Delta u_c|| (P_a))^2}$$

où c sont les composantes du déplacement. Même si le groupe de nœud du pilotage est réduit à un seul nœud, il faut quand même utiliser GROUP_NO.

/'ANA_LIM'

Ce mode de pilotage est spécifique au calcul de charge limite (loi NORTON_HOFF) par approche cinématique (cf. [R7.07.01] pour plus de détail). Si F désigne le chargement assemblé piloté, TYPE_CHARGE='FIXE_PIL0', alors la fonction de pilotage s'écrit simplement :

$$P(u) = F \cdot u = 1$$

Excepté pour le calcul de charge limite, cette fonctionnalité ne présente pas d'intérêt *a priori*. Pour ce mode de pilotage, aucun autre mot clé n'est à préciser.

L'utilisation de lois de comportement adoucissantes peut conduire à des snap backs brutaux qui rendent délicat le déroulement du calcul. Les deux modes de pilotage suivants y remédient (Cf. [R5.03.80] pour plus de détail).

/'DEFORMATION'

DEFORMATION garantit qu'au moins un point de Gauss de la structure voit sa déformation évoluer de façon monotone. On vérifie la relation :

$$\text{cmult} . \max_{\text{point de Gauss}} \left(\frac{\dot{\varepsilon}}{||\dot{\varepsilon}||} \cdot \Delta \varepsilon \right) = \Delta t$$

Ce mode de pilotage est valable pour toutes les lois de comportement y compris en grandes déformations SIMO_MIEHE.

/'PRED_ELAS'

PRED_ELAS assure qu'au moins un point de Gauss de la structure sorte du seuil d'élasticité linéarisé

$f_{\text{pred-elas}}$ d'une quantité $\frac{\Delta t}{\text{cmult}}$. On vérifie la relation :

$$\text{cmult} . \max_{\text{point de Gauss}} (f_{\text{pred-elas}}) = \Delta t$$

Ce mode de pilotage est valable uniquement pour les lois ENDO_FRAGILE (avec la version locale et la version non locale GRAD_EPSI), ENDO_SCALAIRE (avec la version non locale), ENDO_ISOT_BETON et ENDO_ORTH_BETON (avec la version locale et la version non locale), BARENBLATT, BETON_DOUBLE_DP, CZM_EXP (avec les éléments à discontinuité interne *_ELDI), CZM_OUV_MIX et CZM_TAC_MIX (éléments d'interface *_INTERFACE), CZM_EXP_REG (éléments de joint *_JOINT ou modélisation X-FEM) et CZM_LIN_REG (éléments de joint).

La fixation du paramètre cmult est difficile à définir du premier coup parce que la notion de sortie de critère $\frac{\Delta t}{\text{cmult}}$ n'est pas intuitive et varie selon les lois de comportement. Pour les lois ENDO_FRAGILE, ENDO_SCALAIRE et ENDO_ISOT_BETON, une version différente de la définition de $\frac{\Delta t}{\text{cmult}}$ est utilisée, où ce paramètre est lié à l'incrément d'endommagement (voir [R7.01.04]).

Utilisation – Attention :

Lorsqu'on veut utiliser ces deux derniers modes de pilotage, il est indispensable de faire un premier STAT_NON_LINE sans le mot clé PILOTAGE pour amorcer le problème et obtenir un état

initial ε^- différent de zéro (sinon division par zéro pour le pilotage par incrément de déformation). On effectue après une reprise à partir de cet état initial non nul et on utilise le pilotage. De plus, la résolution des deux équations précédentes permet d'obtenir l'intensité du chargement inconnue. Dans certains cas, la résolution de ces équations peut conduire à plusieurs solutions pour l'intensité. On choisit alors toujours la solution qui est la plus proche de ε^- . C'est pourquoi, lorsqu'on veut imposer un chargement alterné, on est obligé à chaque changement de signe du chargement de réaliser un premier STAT_NON_LINE sans le mot clé PILOTAGE afin d'obtenir un état initial ε^- de traction ou de compression. On effectue ensuite un second STAT_NON_LINE en poursuite à partir de l'état initial précédent avec le mot clé PILOTAGE.

Remarque :

| *DEFORMATION et PRED_ELAS ne sont pas disponibles pour les éléments de structures.*

3.11.2 Opérandes NOEUD/GROUP_NO

```
/ NOEUD      = no  
/ GROUP_NO   = grno
```

À n'utiliser qu'avec 'DDL_IMPO', 'LONG_ARC', 'SAUT_IMPO' ou 'SAUT_LONG_ARC'. Pour 'DDL_IMPO', si on utilise l'opérande GROUP_NO, le groupe de nœuds en question ne doit contenir qu'un seul nœud. Dans les autres cas, on utilise uniquement GROUP_NO (qui peut éventuellement ne contenir qu'un seul nœud). Pour 'SAUT_IMPO' et 'SAUT_LONG_ARC', l'opérande est facultative. Pour 'DDL_IMPO' et 'LONG_ARC', on donne le nom du nœud ou le nom du groupe de nœuds sur lequel on va imposer le pilotage.

Pour 'SAUT_IMPO' et 'SAUT_LONG_ARC', la définition est plus subtile puisqu'en modélisation X-FEM on ne pilote pas les valeurs sur des nœuds mais sur des points d'intersection entre les arêtes du maillage et la fissure. Dans la suite, on désigne simplement par « arêtes » les arêtes intersectées. L'algorithme commence par construire un ensemble d'arêtes indépendantes qui couvre toute la fissure (voir fig.3.11.2-1). Par défaut, il pilote sur toutes ces arêtes. Le mot-clé GROUP_NO permet à l'utilisateur de restreindre cet ensemble, chaque nœud renseigné correspondant alors à l'extrémité d'une arête que l'on souhaite piloter. Signalons alors les règles suivantes :

- si deux nœuds sont les extrémités respectives de deux arêtes non indépendantes, une seule sera retenue (fig. 3.11.2-2),
- si un nœud est extrémité de plusieurs arêtes, on retient arbitrairement la première rencontrée par l'algorithme,
- si deux nœuds sont extrémités d'une même fissure (fig.3.11.2-3) une erreur sera renvoyée. D'une façon générale il est conseillé que tous les nœuds entrés soient du même côté de la fissure;
- si un nœud ne correspond à aucune arête (fig. 3.11.2-4), une erreur est renvoyée.

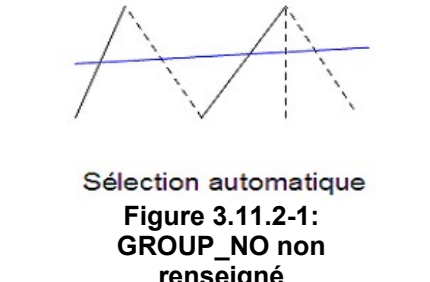
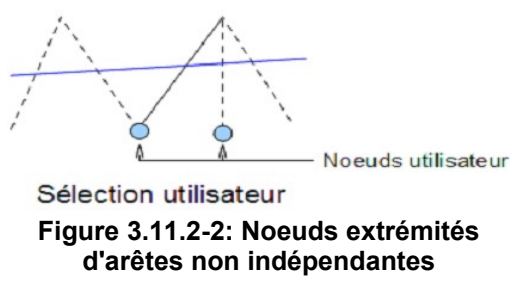
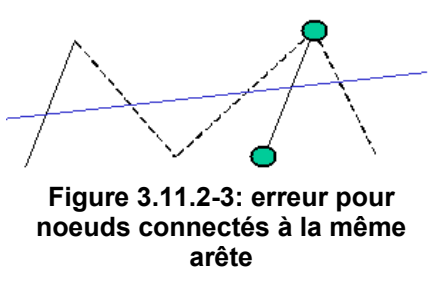
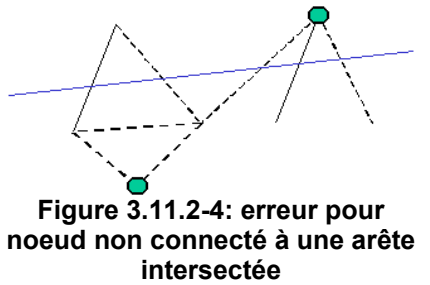
 <p>Sélection automatique Figure 3.11.2-1: GROUP_NO non renseigné</p>	 <p>Sélection utilisateur Figure 3.11.2-2: Noeuds extrémités d'arêtes non indépendantes</p>
 <p>Figure 3.11.2-3: erreur pour noeuds connectés à la même arête</p>	 <p>Figure 3.11.2-4: erreur pour noeud non connecté à une arête intersectée</p>

Tableau 3.1.

3.11.3 Opérandes TOUT/MAILLE/GROUP_MA

```
/ TOUT          = 'OUI'          [DEFAULT]
/ GROUP_MA     = lgrma
/ MAILLE       = lma
```

On donne les mailles ou groupes de mailles servant à piloter le calcul. À n'utiliser qu'avec DEFORMATION ou PRED_ELAS. Intéressant pour alléger la résolution des équations de ces trois modes de pilotages.

3.11.4 Opérande NOM_CMP

```
◇ NOM_CMP = nomcmp
```

C'est le nom de la composante (correspondant au degré de liberté i) utilisée pour le pilotage ('DX' par exemple). À n'utiliser qu'avec 'DDL_IMPO' ou 'LONG_ARC'.

3.11.5 Opérande DIRE_PILO

```
◇ DIRE_PILO = direpilo
```

C'est le nom de la direction i selon laquelle on contrôle le saut de déplacement. Les valeurs possibles sont : DX, DY, DZ, DNOR pour la normale à la fissure, 'DTAN' pour la première tangente (produit vectoriel de la normale avec X), 'DTAN2' pour la deuxième tangente. À n'utiliser qu'avec une modélisation X-FEM. Utilisation pour les types 'SAUT_IMPO', 'SAUT_LONG_ARC' ou avec 'PRED_ELAS' si la sélection sur le choix de la solution pilotée est 'ANGL_INCR_DEPL' ou 'NORM_INCR_DEPL'.

3.11.6 Opérande FISSURE

```
◇ FISSURE = fiss
```

Nom de la `sd_fiss_xfem`. À n'utiliser qu'avec une modélisation X-FEM. Utilisation pour les types 'SAUT_IMPO', 'SAUT_LONG_ARC' ou avec 'PRED_ELAS' si la sélection sur le choix de la solution pilotée est 'ANGL_INCR_DEPL' ou 'NORM_INCR_DEPL'.

3.11.7 Opérande COEF_MULT

```
◇ COEF_MULT = cmult
```


C'est la valeur (notée c_{mult} dans la formule de définition) par laquelle on multiplie le degré de liberté utilisé pour le pilotage. Par défaut, cette valeur vaut 1. A ne pas utiliser avec ANA_LIM.

Exemple avec DDL_IMPO :

Supposons que l'on veut connaître la charge limite d'une structure.

Le chargement imposé sur la structure est la pression d'intensité inconnue ($P = \eta \times$ valeur de référence P_x) sur le groupe de maille A . Pour trouver la charge limite P_{limite} , on va piloter le déplacement du nœud NOI . On veut que le déplacement final suivant x de ce nœud soit égal à 2. (soit d'après la liste d'instantanés des pas de 0.2, soit un coefficient $cmult = 1/0.2 = 5$.)

```
PRESSION = AFFE_CHAR_MECA(PRES = ( GROUP_MA = A, PX = 1.0)),
LIST      = DEFI_LIST_REEL(DEBUT = 0.,
                           INTERVALLE = _F(JUSQU'A = 10, NOMBRE = 10),
RESU      = STAT_NON_LINE(EXCIT = _F( CHARGE = PRESSION,
                                       TYPE_CHARGE = 'FIXE_PILO'),
                           PILOTAGE = _F( TYPE = 'DDL_IMPO',
                                           NOEUD = NOI,
                                           NOM_CMP = 'DX',
                                           COEF_MULT = 5.))
```

Dans le fichier `.resu`, la valeur de η sera affichée à chaque instant du calcul. Pour connaître la charge limite, il suffit de faire $P_{limite} = \eta \times P_x$. (Ici P_x vaut 1 donc on a directement la charge limite). Si on impose sur la structure une pression P proche de la charge limite sans utiliser le pilotage, le calcul ne convergera pas si on est proche de la charge limite.

Attention à la signification de COEF_MULT pour le pilotage de type PRED_ELAS.

3.11.8 Opérande ETA_PILO_R_MAX / ETA_PILO_R_MIN

```
◇ ETA_PILO_R_MAX = etarmax, [R]
◇ ETA_PILO_R_MIN = etarmin, [R]
```

Ces deux mots-clés permettent de définir l'intervalle de recherche des valeurs de pilotage. A chaque itération de Newton toutes les valeurs de pilotage en dehors de $[etarmin, etarmax]$ sont ignorées. Ceci peut emmener à « échec de pilotage » si cet intervalle est trop restrictif.

Si on ne précise pas de valeurs, c'est $-\infty$ pour `etarmin` et $+\infty$ pour `etarmax`. Une utilisation possible de cet intervalle est la suivante. On désire, par exemple, piloter une pression imposée à la structure et on s'attend à garder cette pression positive. En fixant `etarmin` à 0, cela permet d'imposer les valeurs de pilotage positives.

3.11.9 Opérande ETA_PILO_MAX/ETA_PILO_MIN

```
◇ ETA_PILO_MAX = etamax, [R]
◇ ETA_PILO_MIN = etamin, [R]
```

Ces deux mots-clés permettent de préciser l'intervalle de valeurs de pilotage souhaité. On l'utilise pour arrêter proprement le calcul lorsque `ETA_PILOTAGE` atteint les bornes de cet intervalle. Cet intervalle doit être plus restrictif que l'intervalle de recherche défini précédemment, car ce dernier est appliqué dans tous les cas. Le principe de fonctionnement est le suivant : à convergence des itérations de Newton, si l'on a atteint l'une des bornes, le calcul s'arrête. Une utilisation possible de cet intervalle est la suivante. Dans le cas de présence de snap-back en fixant `etamin` à une faible valeur, cela permet d'arrêter le calcul avant une déchirure/endommagement complet de l'échantillon et éviter ainsi la divergence au dernier pas de temps. L'autre utilisation possible est celle de `etamax` en tant que charge limite maximale.

Attention :

Avec la loi `ENDO_ISOT_BETON` et `ENDO_ORTH_BETON` ces deux mots clés sont obligatoires, car ils sont utilisés pour fixer les bornes de pilotage au niveau élémentaire.

3.11.10 Opérateur PROJ_BORNES

```
◇ PROJ_BORNES = /'OUI' [DEFAULT]
                /'NON'
```

En cas de dépassement de l'intervalle (`etamin`, `etamax`), l'utilisateur peut indiquer s'il veut projeter la valeur de pilotage sur (`etamin`, `etamax`).

Avec `PROJ_BORNE='OUI'`, la projection sera effectuée (si `eta>etamax` alors `eta=etamax`; si `eta<etamin` alors `eta=etamin`), ce qui permet, en cas de convergence d'arrêter le calcul précisément sur `etamin` ou `etamax`.

Avec `PROJ_BORNE='NON'`, on ne modifie pas les valeurs de `eta`, même si pendant les itérations de Newton cette dernière a une valeur supérieure à `etamax` ou inférieure à `etamin`. Par contre le calcul est arrêté, si à la convergence `eta` dépassent les bornes.

Une utilisation possible de l'intervalle (`etamin`, `etamax`) avec l'option `PROJ_BORNE='OUI'` est le suivant. On désire, par exemple, comparer plusieurs calcul pour un modèle adoucissant, qui sont piloter en déplacement. Ces paramètres de pilotage permettent de stopper les calculs au même chargement lorsque la structure est suffisamment adoucie. Cette stratégie rend la comparaison plus aisée, grâce au contrôle du dernier point de pilotage.

Avec `PROJ_BORNE='NON'` on arrive dans certains cas à débloquer les calculs, qui autrement ne convergent pas avec les conditions trop restrictives imposées via (`etarmin`, `etamax`). Soit on pilote une pression imposée à la structure et on s'attend à garder cette pression positive. En fixant `etarmin` à 0 le calcul s'arrête en échec de pilotage. En revanche en imposant `etarmin` légèrement négatif, on autorise de facto le passage par un état « non physique » pendant les itérations de Newton, ce qui facilite la convergence. L'état convergé dans ce cas peut aussi bien être physique (pression positive) ou non physique. C'est la valeur de `etarmin=0`, qui gouvernera le comportement en cas de convergence hors borne. Cette stratégie permet de ne conserver que les valeurs de pilotage positives, si on trouve au moins une valeur de pilotage positive.

3.11.11 Opérateur SELECTION

```
◇ /SELECTION = /'NORM_INCR_DEPL', [DEFAULT]
               /'ANGL_INCR_DEPL',
               /'RESIDU',
```

Cet opérateur permet de sélectionner la méthode permettant de choisir la valeur de pilotage dans le cas où plusieurs solutions sont fournies par la résolution de pilotage.

'NORM_INCR_DEPL' permet de sélectionner la valeur de pilotage par la plus petite norme de l'incrément de déplacement sur le pas de temps considéré.

'ANGL_INCR_DEPL' permet de sélectionner la valeur de pilotage par le plus petit angle entre le déplacement obtenu pour le pas de temps courant et le déplacement obtenu pour le pas de temps précédent.

'RESIDU' permet de sélectionner la valeur de pilotage conduisant au plus petit résidu.

Remarque :

Si on fait une reprise de calcul (*reuse*) avec le mot-clé `SELECTION='ANGL_INCR_DEPL'`, il est important de garder à l'esprit que ce critère nécessite les **deux** pas de temps précédents. Il faudra donc bien prendre soin d'archiver correctement les résultats du précédent calcul au risque d'obtenir des résultats faux. Une alarme avertit l'utilisateur.

3.12 Mot clé SOLVEUR

```
◇ SOLVEUR = _F()
```

La syntaxe de ce mot clé commun à plusieurs commandes est décrite dans le document [U4.50.01].

3.13 Mot clé CONVERGENCE

◇ CONVERGENCE = _F ()

Si aucun des deux opérandes suivants n'est présent, alors tout se passe comme si :
RESI_GLOB_RELA = 1.E-6.

3.13.1 Opérande RESI_GLOB_RELA/RESI_GLOB_MAXI

◇ |RESI_GLOB_RELA = resrel , [R]

L'algorithme continue les itérations globales tant que :

$$\max_{i=1,\dots,nbddl} |\mathbf{F}_i^n| > \text{resrel} \cdot \max |\mathbf{L}|$$

où \mathbf{F}^n est le résidu de l'itération n et \mathbf{L} le vecteur du chargement imposé et des réactions d'appuis (Cf. [R5.03.01] pour plus de détails).

Lorsque le chargement et les réactions d'appui deviennent nuls, c'est-à-dire lorsque \mathbf{L} est nul (par exemple dans le cas d'une décharge totale), on passe du critère de convergence relatif au critère de convergence absolu RESI_GLOB_MAXI. Cette opération est transparente pour l'utilisateur (message d'alarme émis dans le fichier .mess). Lorsque le vecteur \mathbf{L} redevient différent de zéro, on repasse automatiquement au critère de convergence relatif RESI_GLOB_RELA.

Si cet opérande est absent, le test est effectué avec la valeur par défaut, sauf si RESI_GLOB_MAXI est présent.

◇ |RESI_GLOB_MAXI = resmax , [R]

L'algorithme continue les itérations globales tant que :

$$\max_{i=1,\dots,nbddl} |\mathbf{F}_i^n| > \text{resmax}$$

où \mathbf{F}^n est le résidu de l'itération n (confer [R5.03.01] pour plus de détails). Si cet opérande est absent, le test n'est pas effectué.

Si RESI_GLOB_RELA et RESI_GLOB_MAXI sont présents tous les deux, les deux tests sont effectués.

Remarque:

Si les conditions limites de Dirichlet sont imposés par AFFE_CHAR_CINE (élimination) et non par AFFE_CHAR_MECA (dualisation), les degrés de libertés portant ces conditions sont ignorées lors de l'évaluation du résidu d'équilibre. Ce qui ne provoque pas de résultats faux mais lorsque le chargement devient nul, c'est-à-dire lorsque \mathbf{L} est nul (par exemple dans le cas d'une décharge totale), on passe du critère de convergence relatif au critère de convergence absolu RESI_GLOB_MAXI. Cette opération est transparente pour l'utilisateur (message d'alarme émis dans le fichier .mess). Lorsque le vecteur \mathbf{L} redevient différent de zéro, on repasse automatiquement au critère de convergence relatif RESI_GLOB_RELA.

3.13.2 Opérande RESI_COMP_RELA

◇ |RESI_COMP_RELA = rescmp , [R]

Cet opérande conduit à estimer la convergence de l'algorithme de Newton en raisonnant composante par composante. Pour cela, on distingue dans le vecteur résidu les sous-vecteurs correspondant à chaque composante *cmp* (par exemple en THM, $\text{cmp} = [DX, DY, DZ, PRE1, PRE2, TEMP]$). On norme ensuite ces sous-vecteurs par la force interne correspondante. Ainsi, l'algorithme continue les itérations globales tant que :

$$\max_{c=1,\dots,nbcmp} \left(\frac{\max_{i=1,\dots,nbddl} |F_i^{c,n}|}{\max_{i=1,\dots,nbddl} |L_i^{\text{int},c,n}|} \right) > \text{rescmp}$$

où $F^{c,n}$ est la partie du résidu F^n correspondant à la composante c et $L^{\text{int},c,n}$ le vecteur des forces internes au temps n correspondant à cette même composante c (Cf. [R5.03.01] pour plus de détails).

Les forces internes au temps n sont calculées en début de pas de temps à partir du résultat issu du pas de temps précédent. Pour le premier pas de temps, on passe automatiquement à un critère relatif de type `RESI_GLOB_RELA`, voir à un critère absolu pour les cas où le chargement est nul. Ce choix n'a d'intérêt que pour des problèmes de type évolutif (THM) où résident de forts contrastes entre les différentes inconnues.

3.13.3 Opérateur RESI_REFE_RELA

	RESI_REFE_RELA	=	resref,	[R]
	SIGM_REFE	=	sigref,	[R]
	FORC_REFE	=	forref,	[R]
	VARI_REFE	=	varref,	[R]
	EPSI_REFE	=	epsref,	[R]
	FLUX_THER_REFE	=	fthref,	[R]
	FLUX_HYD1_REFE	=	fh1ref,	[R]
	FLUX_HYD2_REFE	=	fh2ref,	[R]
	DEPL_REFE	=	depref,	[R]
	LAGR_REFE	=	lagref,	[R]

Cet opérateur conduit à estimer la convergence de l'algorithme de Newton de la manière suivante. A partir d'une référence, qui peut être :

- Une contrainte `sigref` ;
- Une déformation `epsref` ;
- Une variable interne `varref` si l'on utilise des lois non locales à gradient de déformation ;
- Un flux thermique `fthref` dans un cas THM ;
- Deux flux hydriques `fh1ref` et `fh2ref` dans un cas HHM ;
- Un déplacement `depref` si on utilise des éléments de joint avec un comportement de type CZM ;
- Une force ou un moment généralisé `forref` si on utilise des éléments de structure (discrets, barres, poutres ou câbles) ;
- Un coefficient de Lagrange `lagref` pour les formulations mixtes en endommagement.

On calcule une référence de résidu F^{ref} (un vecteur de même longueur que le vecteur résidu). La convergence sera réalisée si et seulement si :

$$\forall i \in [1, \dots, nbddl] \quad |F_i^n| < \text{resref} \cdot F_i^{\text{ref}}$$

3.13.4 Opérateur ITER_GLOB_MAXI

◇ ITER_GLOB_MAXI = /10 [DEFAULT]
/maglob

Nombre d'itérations maximum effectué pour résoudre le problème global à chaque instant (10 par défaut). Ce test est toujours effectué sauf dans le cas du redécoupage du pas de temps par la méthode 'EXTRAPOLE'. L'augmentation excessive de ce paramètre est généralement le signe d'un problème dans la modélisation ou d'une discrétisation temporelle inadéquate.

3.13.5 Opérateur ITER_GLOB_ELAS

◇ ITER_GLOB_ELAS = /25 [DEFAULT]
/maxelas

Nombre d'itérations maximum effectué avec la matrice élastique lorsqu'on utilise le mot clé `PAS_MINI_ELAS` du mot clé `facteur NEWTON` (voir §17) pour résoudre le problème global à chaque instant (25 par défaut).

On rappelle que PAS_MINI_ELAS permet de passer de la matrice tangente à la matrice élastique lorsque le pas de temps est ou devient (par le redécoupage) inférieur à une certaine valeur précisée sous PAS_MINI_ELAS.

Contrairement à ITER_GLOB_MAXI, ce paramètre peut facilement prendre des grandes valeurs (plusieurs centaines) car la convergence sur un problème non-linéaire avec la matrice élastique (très raide) est lente bien qu'assurée du point de vue théorique pour toutes les lois décrivant les matériaux standards généralisés.

3.13.6 Opérandes TYPE/PLATEAU_ITER/PLATEAU_RELA

◇ TYPE	=	/'PIC'	[DEFAULT]
		/'PLATEAU'	
◇ PLATEAU_ITER	=	/3	[DEFAULT]
		/plaite	[I]
◇ PLATEAU_RELA	=	/1E-3	[DEFAULT]
		/plarel	[R]

Cet opérande permet de contrôler le type d'opérateur de convergence à appliquer pour les mots-clefs RESI_*. Par défaut, le mode 'PIC' assure qu'il y a convergence dès lors que la valeur seuil donnée par RESI_* est atteinte. Le mode 'PLATEAU' est plus sévère et exige que le critère de convergence soit stable pendant plaite itérations autour dans un tunnel de largeur plarel autour de la valeur de référence donnée par RESI_*. Ce mode est utile dans les modélisations THM pour lesquelles le critère de Newton est parfois pas assez sévère.

3.13.7 Opérande ARRET

◇ ARRET = /'OUI' [DEFAULT]

Si un des critères de convergence globale choisis n'est pas vérifié après maglob itérations, alors le programme s'arrête (les résultats précédents sont sauvegardés).

◇ ARRET = /'NON' [DEFAULT]

Si maglob est insuffisant pour vérifier les critères de convergence donnés par l'utilisateur, on passe quand même à l'instant suivant. Cette option n'est utilisable qu'en mode DEPL_CALCULE.

Cette option est à utiliser avec précaution car elle donne des résultats faux.

3.14 Mot-clé CRIT_FLAMB

◇ CRIT_FLAMB = _F()

Ce mot-clé permet de déclencher le calcul, à la fin de chaque incrément de temps, d'un critère de stabilité. Ce critère est utile pour déceler, au cours du chargement, le point à partir duquel on perd la stabilité (par flambage par exemple).

Ce critère est calculé de la façon suivante : à la fin d'un pas de temps, en petites perturbations, on résout $\det(\mathbf{K}^T - \lambda \cdot \mathbf{K}^g) = 0$. \mathbf{K}^T est la matrice tangente cohérente à cet instant. \mathbf{K}^g est la matrice de rigidité géométrique, calculée à partir du champ de contraintes à cet instant.

En pratique, le chargement est instable si $|\lambda| < 1$ (en fait $-1 < \lambda < 0$). On calcule les valeurs propres par la méthode de Sorensen (cf. MODE_ITER_SIMULT [U4.52.03]). Ceci peut être assez coûteux pour les problèmes de grande taille.

Pour les grands déplacements et les grandes déformations, on résout $\det(\mathbf{K}^T - \lambda \cdot \mathbf{I}) = 0$ car \mathbf{K}^T contient alors \mathbf{K}^g .

Le critère est alors un critère d'instabilité : quand λ change de signe (donc passe par 0) le chargement est instable. On stocke le mode propre correspondant à la plus petite charge critique (en valeur absolue) dans l'objet résultat, sous le nom MODE_FLAMB. Ce mode propre peut être extrait et visualisé (comme un champ de déplacements ou un mode propre classique). Il est normalisé à 1 sur la plus grande composante de déplacement. L'analyse de stabilité linéaire ne permettant pas de tenir compte de l'aspect suiveur de certaines forces, il faut alors utiliser CRIT_FLAMB.

La documentation [U2.08.04] présentent les différentes approches pour les analyses de stabilité dans Code_Aster, donc CRIT_FLAMB.

3.14.1 Opérande LIST_INST / INST / PAS_CALC

```
◇ /'LIST_INST' = list_r8  
  /'INST'      = l_r8  
  /'PAS_CALC'  = npas
```

Les instants pour lesquels on veut faire un calcul de stabilité sont donnés par une liste d'instant (list_r8 ou l_r8) ou par une fréquence PAS_CALC (tous les npas de temps).

En l'absence de ces mots clés le critère est calculé à tous les pas de temps.

3.14.2 Opérande PRECISION/CRITERE

```
◇ PRECISION = /1.e-6 [DEFAULT]  
              /prec  
◇ CRITERE   = /'RELATIF', [DEFAULT]  
              /'ABSOLU',
```

Permet de sélectionner les instants, c.f. [U4.71.00]

3.14.3 Opérande CHAR_CRIT

```
◇ CHAR_CRIT = /(-10,10), [DEFAULT]  
              /intcc
```

Le mot-clé CHAR_CRIT permet de gagner du temps en ne faisant qu'un test de Sturm dans la bande de fréquence fournie. Si on trouve au moins une fréquence, alors on calcule réellement les valeurs des charges critiques dans cet intervalle.

3.14.4 Opérande NB_FREQ

```
◇ NB_FREQ = / 3 , [DEFAULT]  
            / nbfreq
```

Le mot-clé NB_FREQ (3 par défaut) désigne le nombre de charges critiques à calculer. En fait seule la première suffit mais il peut y avoir des modes multiples.

3.14.5 Opérande RIGI_GEOM

```
◇ RIGI_GEOM = / 'OUI', [DEFAULT]  
              / 'NON'
```

Le mot-clé RIGI_GEOM ('OUI' par défaut) donne le choix à l'utilisateur entre effectuer une recherche de valeurs propres généralisées avec la matrice géométrique au second membre ou non (cas des grandes déformations). Choisir 'NON' signifie que la matrice de raideur géométrique est remplacée par l'identité.

3.14.6 Opérande DDL_EXCLUS

```
◇ DDL_EXCLUS = ('DX', 'DY', ...)
```

Le mot-clé DDL_EXCLUS (liste vide par défaut) désigne l'ensemble des degrés de liberté que l'on souhaite mettre à 0 dans le second membre de la recherche de valeurs propres généralisées. Il ne peut être utilisé que sous la condition RIGI_GEOM='NON'. Cela permet d'imposer des conditions supplémentaires de compatibilité sur les modes propres et ainsi d'effectuer une recherche sélective. Cela est particulièrement adapté aux formulations mixtes. Dans ce cas, l'élimination des multiplicateurs de Lagrange au second membre, permet d'exclure les modes parasites à dominantes Lagrangiennes et de valeurs propres négatives.

3.15 Mot-clé SENSIBILITE

La syntaxe de ce mot clé commun à plusieurs commandes est décrite dans le document [U4.50.02].

3.16 Mot clé ARCHIVAGE

◇ ARCHIVAGE = _F()

Permet d'archiver des ou certains résultats à tous ou certains instants du calcul.

En l'absence de ce mot clé tous les pas de temps sont archivés, y compris les instants de calculs nouvellement créés par redécoupage automatique du pas de temps. L'archivage permet de réduire sensiblement la taille des bases en sélectionnant les instants sauvegardés.

3.16.1 Opérande LIST_INST / INST / PAS_ARCH

◇ /'LIST_INST' = list_r8
/ 'INST' = l_r8
/ 'PAS_ARCH' = npas

La désignation des instants à stocker est effectuée soit par une liste d'instants (list_r8 ou l_r8) ou alors par une fréquence d'archivage (tous les npas de temps).

En l'absence de ces mots clés tous les pas de temps sont archivés.

Remarques :

- Le dernier pas de calcul est toujours stocké pour pouvoir effectuer une reprise ;
- Si on emploie un accès par liste d'instants, alors les instants de calculs nouvellement créés par redécoupage automatique du pas de temps ne sont pas archivés ;
- L'état initial est systématiquement archivé sous le numéro d'ordre 0 dès lors que l'on n'est pas en reprise de calcul (pas de reuse)

3.16.2 Opérande PRECISION/CRITERE

◇ PRECISION = /1.e-6 [DEFAULT]
/prec
◇ CRITERE = /'RELATIF', [DEFAULT]
/'ABSOLU',

Cf. [U4.71.00]

3.16.3 Opérande DETR_NUME_SUIV

DETR_NUME_SUIV = /'NON' [DEFAULT]
/'OUI'

Cette opération peut conduire à écraser des numéros d'ordre préexistants : le mot clé DETR_NUME_SUIV confirme cette destruction, tandis que son absence met fin au calcul.

Illustration

```
LIST = DEFI_LIST_REEL( DEBUT =0.,  
                      INTERVALLE =_F(JUSQU'A =10., NOMBRE =5)),  
  
U2 = STAT_NON_LINE( INCREMENT =_F(LIST_INST =LIST)) ,  
  
U2 = STAT_NON_LINE( reuse=U2,  
                  ETAT_INIT =_F( EVOL_NOLI =U2,  
                                INST =4.),  
                  INCREMENT =_F( LIST_INST =LIST),  
                  ARCHIVAGE =_F( LIST_INST =LIST,  
                                DETR_NUME_SUIV ='OUI')) ,
```

Le résultat de l'archivage pour le premier U2 est le suivant :

numéro d'archivage : 1 2 3 4 5

instants correspondants : 2. 4. 6. 8. 10.

Quand on reprend le calcul (deuxième STAT_NON_LINE), l'instant de reprise (4.0) est inférieur au dernier instant archivé (10.0). Le code émet donc une erreur fatale pour avertir l'utilisateur qu'il risque de perdre des données archivées précédemment (correspondant aux instants 6.0, 8.0 et 10.0). Pour lever cette erreur fatale, il suffit de préciser DETR_NUME_SUIV = 'OUI' .

Le résultat final de l'archivage pour U2 est alors le suivant (les numéros d'ordre 3, 4 et 5 sont recalculés et archivés):

numéro d'archivage	: 1	2	3	4	5
instants correspondants	: 2.	4.	6.	8.	10.

3.16.4 Opérande CHAM_EXCLU

◇ CHAM_EXCLU

Permet de préciser les champs qui ne seront pas archivés, excepté au dernier pas de temps. Le nom des champs exclus dépend des opérateurs.

3.17 Mot clé AFFICHAGE

◇ AFFICHAGE = _F()

Ce mot-clé facteur permet de personnaliser l'affichage du tableau de convergence dans STAT_NON_LINE et DYNA_NON_LINE.

Si ce mot-clé n'est pas renseigné, le tableau est construit suivant les différentes options de calcul (recherche linéaire, pilotage, contact, etc.) et avec INFO_RESIDU = 'NON'.

3.17.1 Opérande UNITE

◇ UNITE = unit

Le tableau de convergence sera dupliqué dans le fichier d'unité *unit*, au format .csv (le séparateur étant la virgule).

3.17.2 Opérande INFO_RESIDU

◇ INFO_RESIDU = / 'NON', [DEFAULT]
/ 'OUI'

Cet opérande permet d'ajouter une colonne pour chaque résidu évalué (RESI_GLOB_RELA, RESI_GLOB_MAXI, RESI_COMP_RELA et RESI_REFE_RELA). Cette colonne indiquera le nœud où le résidu est maximum, ce qui peut aider l'utilisateur lorsqu'il y a des difficultés de convergence. Par exemple, pour voir si le matériau a été mal défini avec une valeur incorrecte sur un élément.

3.17.3 Opérande INFO_TEMPS

◇ INFO_TEMPS = / 'NON', [DEFAULT]
/ 'OUI'

Cet opérande permet d'ajouter une colonne qui donne le temps total passé dans une itération de Newton.

3.18 Mot clé OBSERVATION

◇ OBSERVATION = _F()

Ce mot clé permet de post-traiter certains champs aux nœuds ou aux éléments sur des parties de modèle à des instants d'une liste (dite d'observation) généralement plus raffinée que la liste des instants archivés définie dans le mot clé ARCHIVAGE [§31] (où on stocke tous les champs sur tout le modèle). Il sert essentiellement à des économies de stockage, mais aussi à évaluer des champs sur des parties réduites du maillage, sans avoir besoin de post-traiter après le calcul. Il est possible, par

exemple, de calculer la norme des contraintes au sens de Von-Mises et de la stocker dans la table d'observation.

Ce mot clé est répétable (moins de 99 occurrences) et permet la création d'une table d'observation de même nom que le concept résultat de STAT_NON_LINE que l'on pourra extraire à l'aide de la commande RECU_TABLE.

Il convient d'être prudent dans l'usage de l'observation. En effet, cette fonctionnalité a été conçue pour calculer quelques points dans la structure et il n'est pas conseillé de faire plusieurs milliers d'observations à chaque pas de temps, pour des raisons de performance.

3.18.1 Opérandes LIST_INST / INST / PAS_OBSE

```
◇ /'LIST_INST' = list_r8
  /'INST'      = l_r8
  /'PAS_OBSE'  = npas
```

Ces opérandes permettent de définir aux choix une liste d'instants d'observation. LIST_INST, INST et PAS_OBSE ont la même signification que les opérandes de même nom servant à définir une liste d'archivage. PAS_OBSE jouant le même rôle que PAS_ARCH dans ARCHIVAGE [§31].

3.18.2 Opérandes PRECISION / CRITERE

```
◇ PRECISION = prec
◇ CRITERE   = /'ABSOLU'
              /'RELATIF'
```

Cf. [U4.71.00] pour la syntaxe détaillée.

Ces paramètres permettent de gérer la précision de la sélection des instants pour l'observation.

3.18.3 Opérandes NOM_CHAM / NOM_CMP

```
◆ NOM_CHAM = nomcham
◆ NOM_CMP  = nomcmp
```

Ces opérandes permettent de définir le champ à post-traiter (NOM_CHAM) ainsi que ses composantes données par leur nom (NOM_CMP). On ne peut définir que 20 composantes maximum par occurrence du mot-clef facteur OBSERVATION.

3.18.4 Opérandes TOUT/NOEUD/GROUP_NOEUD/MAILLE/GROUP_MA

```
◇ /TOUT          = 'NON' [DEFAULT]
                        'OUI'
  /NOEUD         = no [no]
  /GROUP_NO      = grno [grno]
  /MAILLE        = lma [ma]
  /GROUP_MA      = lgrma [grma]
```

Ces opérandes permettent de définir le support géométrique de post-traitement :

- Pour des champs aux nœuds ('DEPL','VITE','ACCE', 'DEPL_ABSOLU', 'VITE_ABSOLU', 'ACCE_ABSOLU', 'VALE_CONT', 'FORC_NODA'), on extrait la liste des noeuds.
 - Pour des champs aux points de Gauss ('SIEF_ELGA','VARI_ELGA'), on extrait la liste des mailles.
- Attention à ne pas utiliser TOUT='OUI' sur des maillages de grand taille !

3.18.5 Observation d'un champ ELGA

```
◆ NOM_CMP      = lnocmp , [l_Kn]
◇ EVAL_CMP     = /'VALE ', [DEFAULT]
                  /'FORMULE'
◇ FORMULE      = form [formule_aster]
```

On commence par choisir les composantes ou la formule entre les composantes :

- Si EVAL_CMP = 'VALE', on extrait simplement la liste des composantes donnée par NOM_CMP.
- Si EVAL_CMP = 'FORMULE', on évalue la formule donnée par le mot-clef-simple FORMULE.

Si on applique une formule sur les composantes, on aura donc une valeur et donc une observation, sinon, on aura autant d'observations que de composantes dans la liste NOM_CMP.

```
◇ EVAL_ELGA      =    / 'VALE',                      [DEFAULT]
                    / 'MIN',
                    / 'MAX',
◇ /POINT          =    pi                              [I]
◇ /SOUS_POINT     =    spi                             [I]
```

Une fois évalué les composantes ou la formule sur les composantes, on peut :

- Extraire ces valeurs sur les points et sous-points d'intégration avec EVAL_ELGA = 'VALE'. Dans ce cas, il faut préciser explicitement le point et le sous-point d'intégration par POINT et SOUS_POINT. Les sous-points d'intégration apparaissent pour les éléments de structures (poutres, plaques, coques, tuyaux, etc.).
- Demander d'extraire le maximum EVAL_ELGA = 'MAX' ou le minimum EVAL_ELGA = 'MIN' sur tous les points et sous-points d'une maille.

Si on demande explicitement un point et un sous-point, on aura autant de réalisations que de points demandés, multiplié par le nombre de composantes demandées. Par contre, si on demande le maximum ou le minimum, il y a aura une seule observation par composante demandée.

```
◇ EVAL_CHAM      =    / 'VALE',                      [DEFAULT]
                    / 'MIN',
                    / 'MAX',
                    / 'MOY',
```

Une fois évalué les composantes (ou la formule sur les composantes), ainsi que le point/sous-point d'extraction, on peut :

- Extraire ces valeurs sur toutes les mailles avec EVAL_CHAM= 'VALE'.
- Demander d'extraire le maximum EVAL_CHAM='MAX', le minimum EVAL_CHAM='MIN' ou la moyenne EVAL_CHAM='MOY'.

Exemple: on souhaite extraire le maximum de la trace du tenseur des contraintes sur le GROUP_MA='TOTO'

```
trace          = FORMULE (VALE='0.333*(SIXX+SIYY+SIZZ)',
                          NOM_PARA=('SIXX','SIYY','SIZZ',));
OBSERVATION    = _F(  NOM_CHAM  ='SIEF_ELGA',
                     GROUP_MA  ='TOTO',
                     EVAL_CHAM ='MAX',
                     NOM_CMP  =('SIXX','SIYY','SIZZ',),
                     EVAL_CMP  ='FORMULE',
                     FORMULE   =trace,
                     EVAL_ELGA ='MAX')
```

3.18.6 Observation d'un champ NOEU

```
◆ NOM_CMP      =    lnocmp ,                      [1_Kn]
◇ EVAL_CMP     =    / 'VALE ',                      [DEFAULT]
                    / 'FORMULE'
◇ FORMULE      =    form                            [formule_aster]
```

On commence par choisir les composantes ou la formule entre les composantes :

- Si EVAL_CMP = 'VALE', on extrait simplement la liste des composantes donnée par NOM_CMP.
- Si EVAL_CMP = 'FORMULE', on évalue la formule donnée par le mot-clef-simple FORMULE.

Si on applique une formule sur les composantes, on aura donc une valeur et donc une observation, sinon, on aura autant d'observations que de composantes dans la liste NOM_CMP.

```

◇ EVAL_CHAM = / 'VALE', [DEFAULT]
               / 'MIN',
               / 'MAX',
               / 'MOY',

```

Une fois évalué les composantes (ou la formule sur les composantes), ainsi que le point/sous-point d'extraction, on peut :

- Extraire ces valeurs sur toutes les mailles avec EVAL_CHAM= 'VALE'.
- Demander d'extraire le maximum EVAL_CHAM= 'MAX', le minimum EVAL_CHAM= 'MIN' ou la moyenne EVAL_CHAM= 'MOY'.

Exemple: on souhaite extraire le maximum de la composante DX du déplacement sur GROUP_NO = 'TOTO'

```

OBSERVATION =_F(  NOM_CHAM = 'DEPL',
                  GROUP_NO = 'TOTO',
                  EVAL_CHAM = 'MAX',
                  NOM_CMP = ('DX',),)

```

3.19 Mot clé SUIVI_DDL

```

◇ SUIVI_DDL =_F()

```

Ce mot clé permet de post-traiter certains champs aux nœuds ou aux éléments sur des parties de modèle à toutes les itérations de Newton et les afficher dans le tableau de convergence.

Le nombre simultanée de SUIVI_DDL dépend des colonnes affichées et donc des fonctionnalités activées.

Le SUIVI_DDL a la même syntaxe que OBSERVATION pour l'extraction des champs, sauf que l'on ne donne pas d'informations sur les instants à extraire, puisqu'on le réalise à chaque itération de Newton (il n'y a pas les mots clefs LIST_INST/INST/PAS_OBSE/CRITERE/PRECISION).

```

◇ TITRE = ltitre, [list_k]

```

Ce mot-clef attend une liste de trois chaînes au maximum et permet de nommer la colonne du tableau d'affichage. Les chaînes sont tronquées à 16 caractères.

3.20 Contenu de la structure de données EVOL_NOLI

La structure de données EVOL_NOLI contient la liste des champs archivés au cours du calcul (selon les différentes options de mot-clef ARCHIVAGE). Par défaut, elle contient, pour chaque instant, la liste des champs suivants :

- DEPL : champ (aux nœuds) des déplacements ;
- SIEF_ELGA : champ (aux points de Gauss) des contraintes ;
- VARI_ELGA : champ (aux points de Gauss) des variables internes ;
- COMPOR : carte du comportement ;

Selon certaines options de calcul, d'autres champs seront présents :

- VALE_CONT : champ (aux nœuds) des informations sur le contact-frottement (voir [U4.44.11] pour plus de détails sur le contenu de ce champ) ;
- INDC_ELEM : champ (aux points de Gauss) des statuts de contact pour le cas XFEM avec contact ;
- COHE_ELEM : champ (aux points de Gauss) du paramètre de cohésion pour le cas XFEM avec RELATION='CZM' ;
- SECO_ELEM : champ (aux points de Gauss) des statuts de frottement pour le cas XFEM avec contact et frottement ;

En plus de ces champs, la structure de données contient également des paramètres. A chaque instant, on stocke au minimum :

Nom	Mot-clef origine	Description	Type
-----	------------------	-------------	------

INST		Valeur de l'instant de calcul	R
EXCIT	EXCIT	Informations sur les chargements	K24
MODELE	MODELE	Modèle	K8
CARAELEM	CARA_ELEM	Caractéristiques élémentaires	K8
CHAMPMAT	CHAM_MATER	Champ de matériau	K8
PARM_THETA	COMP_INCR/PARM_THETA	Paramètre d'intégration de la loi de comportement	R
ITER_GLOB		Nombre total d'itérations de Newton	I
CHAR_MINI		Chargement minimum atteint au cours du pas de temps	R
ETA_PILOTAGE		Paramètre de pilotage	R

Quand on recherche des modes de flambement ou des modes vibratoires, on stocke le champ de déplacement correspondant et la valeur du chargement critique ou la fréquence.

Nom	Mot-clef origine	Description	Type
CHAR_CRIT	CRIT_FLAMB	Chargement critique du mode de flambement	R
FREQ	MODE_VIBR	Fréquence du mode vibratoire	E

3.21 Opérande INFO

◇ INFO = inf

Permet d'effectuer dans le fichier message diverses impressions intermédiaires.

D'autres impressions sont faites systématiquement lors du calcul non linéaire, indépendamment de la valeur affectée au mot-clé INFO : ce sont les impressions des résidus et des incréments relatifs de déplacement au cours des itérations de Newton.

Attention, les fichiers .mess peuvent devenir très importants avec INFO = 2.

3.22 Opérande TITRE

◇ TITRE = tx

tx est le titre du calcul. Il sera imprimé en tête des résultats. Voir [U4.03.01].