

Manuel d'Utilisation
Fascicule U4.5- : Méthode de résolution
Document : U4.51.03

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). 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	4
3 Opérandes	9
3.1 Opérandes MODELE / CHAM_MATER / CARA_ELEM	9
3.2 Mot clé EXCIT	9
3.2.1 Opérandes CHARGE	9
3.2.2 Opérande FONC_MULT	10
3.2.3 Opérande TYPE_CHARGE	10
3.3 Mot-clé SOUS_STRUC	11
3.4 Mots-clés COMP_INCR et COMP_ELAS	11
3.5 Mot clé VARI_COMM	11
3.5.1 Opérande IRRAS & CORROSION	11
3.6 Mot clé ETAT_INIT	11
3.6.1 Opérandes SIGM / VARI / DEPL / VARI_NON_LOCAL	12
3.6.2 Opérandes EVOL_NOLI	12
3.6.3 Opérande NUME_ORDRE / INST / NUME_DIDI	12
3.6.4 Opérande INST_ETAT_INIT	12
3.6.5 Opérande PRECISION / CRITERE	13
3.7 Mot clé INCREMENT	14
3.7.1 Opérandes LIST_INST / EVOLUTION	14
3.7.2 Opérandes NUME_INST_INIT / INST_INIT / NUME_INST_FIN / INST_FIN	14
3.7.3 Opérande PRECISION	15
3.7.4 Opérande SUBD_PAS / SUBD_PAS_MINI / COEF_SUBD_PAS_1	15
3.7.5 Opérande OPTI_LIST_INST / NOM_CHAM / NOM_CMP / VALE	16
3.8 Mot clé NEWTON	16
3.8.1 Opérande PREDICTION	16
3.8.2 Opérande MATRICE	17
3.8.3 Opérande EVOL_NOLI	18
3.9 Mot clé RECH_LINEAIRE	18
3.9.1 Opérande RESI_LINE_RELA / ITER_LINE_MAXI	18
3.9.2 Opérande PAS_MINI_CRIT / ITER_LINE_CRIT	19
3.9.3 Opérandes RHO_MIN / RHO_MAX / RHO_EXCL	19
3.10 Opérande PARM_THETA	19
3.11 Mot clé PILOTAGE	19
3.11.1 Opérande TYPE	20
3.11.2 Opérandes NOEUD / GROUP_NO	21
3.11.3 Opérandes TOUT / MAILLE / GROUP_MA	21

Titre : Opérateur *STAT_NON_LINE*
Auteur(s) : **P. BADEL**

Date : 11/03/05
Clé : **U4.51.03-G** Page : 3/30

3.11.4	Opérande NOM_CMP.....	21
3.11.5	Opérande COEF_MULT	22
3.11.6	Opérande ETA_PILO_R_MAX / ETA_PILO_R_MIN.....	22
3.11.7	Opérande ETA_PILO_MAX / ETA_PILO_MIN.....	22
3.11.8	Opérande PROJ_BORNES	23
3.11.9	Opérande SELECTION	23
3.12	Mot clé SOLVEUR	23
3.13	Mot clé CONVERGENCE	23
3.13.1	Opérande RESI_GLOB_RELA / RESI_GLOB_MAXI.....	23
3.13.2	Opérande RESI_REFE_RELA.....	24
3.13.3	Opérande ITER_GLOB_MAXI	24
3.13.4	Opérande ITER_GLOB_ELAS.....	24
3.13.5	Opérande ARRET.....	24
3.13.6	Opérandes RESI_INTE_RELA / ITER_INTE_MAXI	25
3.13.7	Opérande ITER_INTE_PAS	25
3.13.8	Opérande RESO_INTE	25
3.14	Mot-clé CRIT_FLAMB.....	26
3.15	Mot-clé SENSIBILITE.....	26
3.16	Mot clé ARCHIVAGE	26
3.16.1	Opérande LIST_INST / INST / PAS_ARCH	27
3.16.2	Opérande PRECISION	27
3.16.3	Opérande ARCH_ETAT_INIT / NUME_INIT / DETR_NUME_SUIV.....	27
3.16.4	Opérande CHAM_EXCLU	28
3.17	Opérande OBSERVATION.....	29
3.18	Opérande SOLV_NON_LOCAL	29
3.19	Opérande LAGR_NON_LOCAL	29
3.20	Opérande INFO	30
3.21	Opérande TITRE	30

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_PILO'
                        / 'SUIV'
                        / 'DIDI'
    ),
  ◆ SOUS_STRUC = _F (
    ◆ CAS_CHARGE = chi,                [char_meca]
    ◆ / TOUT      = 'OUI',              [DEFAULT]
    / MAILLE      = lma,                [l_maille]
  ),
  ◆ | COMP_INCR = _F (voir le document [U4.51.11] ),
  | COMP_ELAS = _F (voir le document [U4.51.11] ),
  ◇ VARI_COMM = _F (
    ◆ / IRRA      = irra                [evol_varc]
    ◆ / CORROSION = corro                [evol_varc]
  ),
  ◇ ETAT_INIT = _F (
    ◆ / | SIGM =          sig,           [cham_elem_SIEF_R]
        | VARI =          vain,          [carte_SIEF_R]
        | DEPL =          depl,          [cham_elem_VARI_R]
        | VARI_NON_LOCAL = vanolo,       [cham_no_DEPL_R]
        | VARI_NON_LOCAL = vanolo,       [cham_no_VANL_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 = litps, [listr8]

    ◇ EVOLUTION = / 'CHRONOLOGIQUE', [DEFAULT]
                  / 'RETROGRADE',
                  / 'SANS',

    ◇ / 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]

    ◇ SUBD_PAS = / 1, [DEFAULT]
                 / subpas, [I]

    ◇ SUBD_PAS_MINI = submini, [R]

    ◇ COEF_SUBD_PAS_1 = / 1., [DEFAULT]
                       / coefsub, [R]

    ◇ OPTI_LIST_INST = / 'INCR_MAXI', [DEFAULT]

    ◇ NOM_CHAM = nomch, [Kn]

    ◇ NOM_CMP = nomcmp, [Kn]

    ◇ VALE = val [R]
    ),

◇ 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]
    ),

```

Titre : *Opérateur STAT_NON_LINE*
Auteur(s) : **P. BADEL**

Date : 11/03/05
Clé : *U4.51.03-G* Page : 6/30

```
◇ RECH_LINEAIRE =_F (
    ◇ 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]
    ),
◇ PARM_THETA =                  / 1.,            [DEFAULT]
                                   / theta,           [R]
◇ PILOTAGE =_F (
    ◆ TYPE =                     / 'DDL_IMPO',
                                   / '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 =                grno,             [gr_noeud]

    ◇ NOM_CMP =                  nomcmp,             [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',

    ),
```

Titre : Opérateur *STAT_NON_LINE*
Auteur(s) : **P. BADEL**

Date : 11/03/05
Clé : *U4.51.03-G* Page : 7/30

```
◇ 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_REFE_RELA = resref, [R]  
    ◇ SIGM_REFE = sigref, [R]  
    ◇ EPSI_REFE = sigref, [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]  
  
    ◇ ARRET = / 'OUI', [DEFAULT]  
             / 'NON',  
  
    ◇ RESI_INTE_RELA = / 1.E-6, [DEFAULT]  
                      / resint, [R]  
  
    ◇ ITER_INTE_MAXI = / 10, [DEFAULT]  
                      / iteint, [I]  
  
    ◇ ITER_INTE_PAS = / 0, [DEFAULT]  
                     / itepas, [I]  
  
    ◇ RESO_INTE = / 'IMPLICITE', [DEFAULT]  
                 / 'RUNGE_KUTTA_2',  
                 / 'RUNGE_KUTTA_4',  
    ),  
◇ CRIT_FLAMB =_F (   
    ◇ NB_FREQ = / 3, [DEFAULT]  
               / nbfreq, [I]  
    ◇ CHAR_CRIT = / (-10,10), [DEFAULT]  
                 / intcc,  
    ),  
◇ SENSIBILITE ( voir le document [U4.50.02] ),  
◇ ARCHIVAGE =_F (   
    ◇ / LIST_INST = list_r8, [listr8]  
      / INST = l_r8, [R]  
      / PAS_ARCH = npas, [I]  
  
    ◇ PRECISION = / 1.E-3, [DEFAULT]  
                  / prec, [R]  
  
    ◇ / ARCH_ETAT_INIT = 'OUI',  
      / NUME_INIT = nuinit, [I]  
  
    ◇ DETR_NUME_SUIV = 'OUI',
```

Titre : Opérateur *STAT_NON_LINE*
Auteur(s) : **P. BADEL**

Date : 11/03/05
Clé : U4.51.03-G Page : 8/30

```

      ◇ CHAM_EXCLU =      | 'DEPL',
                           | 'SIEF_ELGA',
                           | 'VARI_ELGA',
                           | 'VARI_NON_LOCAL',
                           | 'LANL_ELGA',
    ),
◇ OBSERVATION =_F ( voir le document [U4.53.01] ),
◇ LAGR_NON_LOCAL =_F (
    ◇ ITER_PRIM_MAXI =      / 10,                [DEFAULT]
                           / iterprimmax,        [I]
    ◆ RESI_PRIM_ABSO =      resiprimab,          [R]
    ◇ ITER_DUAL_MAXI =      / 50,                [DEFAULT]
                           / iterdmax,          [I]
    ◆ RESI_DUAL_ABSO =      residabso,           [R]
    ◇ R =                  / 1000.              [DEFAULT]
                           / rho                [R]
    ),
◇ SOLV_NON_LOCAL =_F ( voir le document [U4.50.01] ),
◇ INFO =                  / 1,                [DEFAULT]
                           / 2,
◇ TITRE = tx              [Kn]
);
```


3 Opérandes

3.1 Opérandes *MODELE* / *CHAM_MATER* / *CARA_ELEM*

- ◆ *MODELE* = *mo*
- ◆ *CHAM_MATER* = *chmat*
- ◇ *CARA_ELEM* = *carac*

Ces mots-clés permettent de renseigner :

- le nom du modèle (*mo*) dont les éléments font l'objet du calcul mécanique,
- le nom du champ de matériau (*chmat*) affecté sur le maillage. Attention, toutes les mailles du modèle doivent être associées à un matériau (sinon erreur fatale avec message peu explicite),
- le nom des caractéristiques (*carac*) des éléments de coque, poutre, tuyau, barre, câble, et éléments discrets affectés sur le modèle *mo*. Evidemment, 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* :

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

3.2.1 Opérandes *CHARGE*

- ◆ *CHARGE* : *ch_i*

ch_i est le chargement mécanique (comportant éventuellement l'évolution d'un champ de température) précisé à la *i*^{ème} occurrence de *EXCIT*.

*Une et une seule charge peut comporter l'évolution d'un champ de température, qui aura précédemment été défini grâce au mot-clé *TEMP_CALCULEE* de la commande *AFFE_CHAR_MECA*.*

Attention :

*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.

Attention :

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 de 1 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 ».

Attention :

| Les chargements issus de *AFFE_CHAR_CINE* ne sont pas utilisables avec *STAT_NON_LINE*.

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

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

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

Par défaut : $f_i = 1$.

Remarque :

| Le champ de température n'est pas multiplié par f_i .

3.2.3 Opérande TYPE_CHARGE

◇ **TYPE_CHARGE** : $tchi$

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_PILLO', le chargement est toujours fixe (indépendant de la géométrie) mais sera piloté grâce au mot clé PILOTAGE [§3.11]. Les charges pilotables doivent être issues d'AFFE_CHAR_MECA ou d'AFFE_CHAR_MECA_F 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 SOUS_STRUC

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]

◆ **SOUS_STRUC**

Ce mot clé facteur permet de préciser quels sont les chargements à utiliser pour les sous-structures. 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.

◆ **CAS_CHARGE = nocas**

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

◆ **/ TOUT = 'OUI'**

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

/ MAILLE = l_mail

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

3.4 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].

3.5 Mot clé VARI_COMM

◇ **VARI_COMM :**

Variables de commandes qui pilotent les lois de comportement (au même titre que la température).

3.5.1 Opérande IRRA & CORROSION

◆ **/ IRRA : irr**

Champs d'irradiation.

◆ **/ CORROSION : corro**

Champs de corrosion.

3.6 Mot clé ETAT_INIT

◇ **ETAT_INIT :**

Etat initial de référence choisi. Par défaut, tous les champs sont identiquement nuls. Cet 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.

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 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.

Remarque :

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.

3.6.1 Opérandes SIGM / VARI / DEPL / VARI_NON_LOCAL

♦ /	SIGM	=	sig
	VARI	=	vain
	DEPL	=	depl
	VARI_NON_LOCAL	=	vano1o

Respectivement, champs de contraintes aux points de Gauss, de variables internes aux points de Gauss, de déplacements aux nœuds et de variables non locales aux nœuds (pour des modèles non locaux) pris à l'état initial. 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 par la commande *LIRE_RESU* (attention le format MED ne lit que des champs aux nœuds).

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 (déplacement) qui sert de référence pour l'application de ces conditions aux limites (Cf. [§3.2.2]). 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 :

- lorsque l'état initial est défini par la donnée des champs, il n'y a pas d'instant associé.
- lorsque l'état est donné par un concept *evol_noli*, il s'agit de l'instant dans le précédent calcul (*istetaini* = *instini*).

A - Exemple simple (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)) ,
```

1^{er} STAT_NON_LINE : effectue le calcul pour les instants 1, 2, 3 et 4s.

2nd 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'instants 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.)) ,
```

1^{er} STAT_NON_LINE : effectue le calcul des instants 1 à 10s.

2nd STAT_NON_LINE : effectue le calcul des instants 21 à 30s, l'état initial correspondant à l'instant t=10s du 1^{er} STAT_NON_LINE (par défaut INST=10.). Cet état initial correspond pour ce 2nd 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.)) ,
```

1^{er} STAT_NON_LINE : effectue le calcul des instants 1 à 10s.

2nd STAT_NON_LINE : effectue le calcul des instants 1 à 10s, l'état initial correspondant à l'instant t=10s du 1^{er} STAT_NON_LINE (par défaut INST=10.). Cet état initial correspond pour ce 2nd STAT_NON_LINE à l'instant t=0s. (INST_ETAT_INIT : 0.).

3.6.5 Opérande PRECISION / CRITERE

Cf. [U4.71.00].

3.7 Mot clé INCREMENT

◆ **INCREMENT :**

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'indicer les incréments de charge et de paramétrer l'évolution d'un éventuel champ de température.

3.7.1 Opérandes LIST_INST / EVOLUTION

◆ **LIST_INST : litps**

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

◇ **EVOLUTION :** / 'CHRONOLOGIQUE' [DEFAULT]
 / 'RETROGRADE'
 / 'SANS'

Le mot clé 'CHRONOLOGIQUE' permet de vérifier si la liste d'instants donnée par l'utilisateur est strictement croissante (si non un message d'erreur est émis).

Le mot clé 'RETROGRADE' permet d'inverser la liste d'instants donnée par l'utilisateur et de vérifier qu'après cette opération, elle est bien strictement décroissante.

Il n'y a pas de vérification lorsqu'on précise une évolution 'SANS'.

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 (`EVOLUTION : 'CHRONOLOGIQUE' ou 'RETROGRADE'`).

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

- 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 l'état initial,
- s'il n'y a pas d'état initial (opérande `ETAT_INIT`) 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 `litps` (`NUME_INST_INIT : 0`), ou le dernier lorsque l'évolution est rétrograde.

◇ / **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 : avec une évolution `RETROGRADE`, `INST_INIT > INST_FIN`.

Titre : Opérateur *STAT_NON_LINE*
Auteur(s) : **P. BADEL**

Date : 11/03/05
Clé : **U4.51.03-G** Page : 15/30

A - Exemple simple (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)) ,
```

1^{er} STAT_NON_LINE : effectue le calcul pour les instants 1, 2, 3 et 4s.

2nd 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)) ,
```

1^{er} STAT_NON_LINE : effectue le calcul des instants 1 à 4s.

2nd 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]

3.7.4 Opérande SUBD_PAS / SUBD_PAS_MINI / COEF_SUBD_PAS_1

◇ **SUBD_PAS** = subpas
◇ **SUBD_PAS_MINI** = submini
◇ **COEF_SUBD_PAS_1** = coefsub

Permet de réaliser un redécoupage automatique du pas de temps lorsque l'algorithme de Newton ne converge pas.

Le pas de temps est redécoupé en subpas sous pas. Par défaut il n'y a pas de redécoupage (subd_pas : 1). La subdivision automatique s'arrête lorsque les nouveaux pas créés sont plus petits que SUBD_PAS_MINI. Les nouveaux pas créés sont de taille identique, excepté le premier qui est égal à cette taille multipliée par COEF_SUBD_PAS_1 (par défaut 1). Ceci permet de mieux prendre en compte les problèmes de décharge de la structure (changement de matrice tangente) sans utiliser la matrice élastique (PREDICTION : 'ELASTIQUE' ou MATRICE : 'ELASTIQUE' sous l'opérateur NEWTON).

Lorsqu'un pas de temps a été redécoupé plusieurs fois (appelons n le nombre de fois où l'on a procédé à une subdivision du même pas), le pas suivant est automatiquement subdivisé (n-1) fois, ceci pour éviter, en cas de convergence difficile de tenter un pas de temps trop important.

Remarque concernant le mot clé DECOUPE sous SOLVEUR :

Lors de calcul de flambage élastoplastique, il peut arriver que la matrice tangente du système soit singulière au cours des itérations de Newton. En redécoupant le pas de temps, on peut passer ces points durs. Sous l'opérateur SOLVEUR, le mot clé DECOUPE sous STOP_SINGULIER sert à gérer ces points durs. Il est alors nécessaire de renseigner les mots clés relatifs au redécoupage pour que la méthode DECOUPE soit activée.

3.7.5 Opérande OPTI_LIST_INST / NOM_CHAM / NOM_CMP / VALE

```

◇ OPTI_LIST_INST = 'INCR_MAXI' [DEFAULT]
◇ NOM_CHAM       = 'TEMP'      [DEFAULT]
◇ NOM_CMP        = 'TEMP'      [DEFAULT]
◇ VALE           = vale

```

Ces opérandes n'ont d'intérêt que lorsqu'on réalise un calcul thermomécanique. Permet de créer si besoin une nouvelle liste de pas de temps mécanique de sorte que, entre chaque incrément de temps, l'incrément de température soit inférieur à une valeur donnée par l'utilisateur et renseignée par le mot clé VALE.

La création de cette nouvelle liste se fait de la façon suivante :

- Liste d'instants initial en mécanique : T_i
- Liste d'instants thermique : θ
- Nouvelle liste d'instants mécanique finale à créer si besoin : T_f
- On insère entre chaque intervalle de la liste initiale mécanique T_i , les instants thermiques inclus dans cet intervalle. On récupère alors pour chaque intervalle une liste d'instants $\tau = [\tau_0, \tau_1, \tau_2, \dots, \tau_N]$
- Construction de la liste final T_f
 - Initialisation : $\tau_f = \tau_0$
 - 1^{er} Test :
 - Si $T(\tau_j) - T(\tau_f) > \text{valeur}$ avec $T(t)$ la température au temps t et τ_f le dernier instant inséré dans la nouvelle liste T_f , alors on garde dans la nouvelle liste T_f , l'instant τ_{j-1}
 - 2^{ème} Test :
 - Si $T(\tau_j) - T(\tau_{j-1}) > \text{valeur}$ alors on redécoupe uniformément cet intervalle de façon à satisfaire la condition sur l'incrément de température.

Exemple : SI $T(\tau) = [T(\tau_1) = 20^\circ\text{C}, T(\tau_2) = 30^\circ\text{C}, T(\tau_3) = 55^\circ\text{C}, T(\tau_4) = 65^\circ\text{C}]$ avec VALE = 15°C

Initialisation : $\tau_f = \tau_1$

Intervalle 1 :

1^{er} Test = 2^{ème} Test : $T(\tau_2) - T(\tau_1) = 10^\circ\text{C} < 15$ donc on $T_f = [\tau_1]$

Intervalle 2 :

1^{er} Test : $T(\tau_3) - T(\tau_f) = 35^\circ\text{C} > 15$ donc on a $T_f = [\tau_1, \tau_2]$ et $\tau_f = \tau_2$

2^{ème} Test : $T(\tau_3) - T(\tau_2) = 25^\circ\text{C} > 15$ donc on $T_f = [\tau_1, \tau_2, T_3]$ tel que $T(T_3) = 42.5^\circ\text{C}$, $\tau_3]$ et $\tau_f = \tau_3$

Intervalle 3 :

1^{er} Test = 2^{ème} Test : $T(\tau_4) - T(\tau_3) = 10^\circ\text{C} < 15^\circ\text{C}$

d'où la liste finale suivante :

$T_f = [\tau_1, \tau_2, T_3 \text{ tel que } T(T_3) = 42.5^\circ\text{C}, \tau_3, \tau_4]$

3.8 Mot clé NEWTON

◇ **NEWTON :**

Précise les caractéristiques de la méthode de résolution du problème incrémental non linéaire (méthode de NEWTON-RAPHSON).

3.8.1 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` dans le fichier `.mess`) 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` dans le fichier `.mess`) qui est utilisée si on a choisi `MATRICE: 'ELASTIQUE'`.

/ **'TANGENTE'**

On utilise la matrice tangente du problème en vitesse (option `RIGI_MECA_TANG` dans le fichier `.mess`).

/ **'ELASTIQUE'**

On utilise la matrice élastique (option `RIGI_MECA` dans le fichier `.mess`).

/ **'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` (§3.8.3).

Utilité :

- 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 calcul 1 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.
- 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.

3.8.2 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] qui est réévaluée tous les `mf` incréments de temps (`mf` positif ou nul) et 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.

Titre : Opérateur *STAT_NON_LINE*
Auteur(s) : **P. BADEL**

Date : 11/03/05
Clé : U4.51.03-G Page : 18/30

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

Permet 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 ou devient (par le redécoupage) 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, prendre `REAC_ITER_ELAS = 0` (puisqu'elle sera la même au cours des itérations).

Utilité :

Cette option peut être utile lorsque le redécoupage automatique du pas de temps (cf. [§ 3.7.4]) 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.

/ '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 de la relation de comportement).

3.8.3 Opérande EVOL_NOLI

```
◇ EVOL_NOLI : evol_noli
```

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

3.9 Mot clé RECH_LINEAIRE

RECH_LINEAIRE :

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 `GREEN_GR` pour les modélisations `COQUE_3D` et en présence de contact.

3.9.1 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.

Il n'est pas nécessaire de spécifier une précision ni un nombre d'itérations très élevés, la pratique montrant que 2 ou 3 itérations de recherche linéaire sont suffisantes. On peut donc se contenter de demander 3 itérations avec la précision par défaut.

3.9.2 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_MAXI)

3.9.3 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 de la recherche linéaire, sous la forme :
 $I = [r \text{ min}, r \text{ max}] - [-rexc, rexc]$.

3.10 Opérande PARM_THETA

```

◇ PARM_THETA = / 1. [DEFAULT]
               / theta
    
```

Pour les modélisations THM, l'argument theta est le paramètre de la thêta-méthode utilisée pour résoudre les équations évolutives de thermique et d'hydraulique (Cf. [R5.03.60] pour plus de détails). Sa valeur doit être comprise entre 0 (méthode explicite) et 1 (méthode totalement implicite).

Pour les lois de comportement ROUSS_VISC, ASSE_COMBU, ZIRC_CYRA2 et ZIRC_EPRI, l'argument theta sert à l'intégration de la loi de comportement (pour le modèle ASSE_COMBU, il sert à intégrer la loi de Lemaitre en 1D). Il peut prendre les valeurs 0.5 (semi-implicite) ou 1 (implicite).

3.11 Mot clé PILOTAGE

◇ PILOTAGE :

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_PILO), 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_PILO, on ne peut pas utiliser pour le chargement de référence le mot clé FONCT_MULT.

Attention :

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.

Attention :

Le mot clé PILOTAGE est interdit avec le contact.

3.11.1 Opérande TYPE

◇ TYPE : / 'DDL_IMPO'
 / 'LONG_ARC'
 / 'ANA_LIM'
 / 'DEFORMATION'
 / 'PRED_ELAS'

C'est le type de pilotage effectué. Cinq 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 :

$$c_{mult} \Delta u_i(no) = \Delta t$$

/ '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 :

$$C_{mult} \|\Delta u\| = \Delta t \text{ avec } \|\Delta u\| = \sqrt{\sum_n \sum_c \Delta U_{n,c}^2}$$

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.

/ '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_PILO', alors la fonction de pilotage s'écrit simplement :

$$P(U) = F.U = 1$$

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.

Remarque :

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 :

$$C_{mult} \max_{\text{point de Gauss}} \left(\frac{\varepsilon^-}{\|\varepsilon^-\|} \Delta \varepsilon \right) = \Delta t$$

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 $f_{\text{préd-élas}}$ d'une quantité $\Delta T/C_{\text{mult}}$. On vérifie la relation :

$$C_{\text{mult}} \max_{\text{point de Gauss}} (f_{\text{préd-élas}}) = \Delta t$$

Ce mode de pilotage est valable uniquement pour les lois ENDO_FRAGILE (avec la version locale et les deux versions non locales), ENDO_ISOT_BETON et ENDO_ORTH_BETON (avec la version locale et la version non locale), BARENBLATT et BETON_DOUBLE_DP.

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

On donne le nom du nœud ou le nom de groupe de nœuds sur lequel on va imposer le pilotage. A n'utiliser qu'avec 'DDL_IMPO' ou '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. Pour 'LONG_ARC', on utilise uniquement GROUP_NO (qui peut éventuellement ne contenir qu'un seul nœud).

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. A 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). A n'utiliser qu'avec 'DDL_IMPO' ou 'LONG_ARC'.

3.11.5 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 veuille connaître la charge limite d'une structure.

Le chargement imposé sur la structure est la pression d'intensité inconnue ($P = \eta \cdot \text{valeur de référence } P_x$) sur le groupe de maille A. Pour trouver la charge limite P_{limite} , on va piloter le déplacement du nœud NO1. On veut que le déplacement final suivant x de ce nœud soit égal à 2. (soit d'après la liste d'instants des pas de 0.2, soit un coefficient $cmult = 1/0.2 = 5$.)

[illegible]

Dans le fichier.resu, la valeur de η sera affichée à chaque instant du calcul. Pour connaître la charge limite, il suffit de faire $P_{\text{limite}} = \eta * P_x$. (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.

3.11.6 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 préciser l'intervalle de valeurs de pilotage attendues. Le principe de fonctionnement est le suivant : à chaque itération de Newton, si l'on trouve des valeurs de pilotage dans l'intervalle $[etar\ min, etar\ max]$, toutes les valeurs de pilotage en dehors de cet intervalle ne sont pas considérées. En revanche, si aucune valeur de pilotage n'est trouvée dans cette intervalle, toutes les valeurs de pilotage sont conservées.

Si on ne précise pas de valeurs, c'est $-\infty$ pour `etamin` et $+\infty$ pour `etamax`.

Une utilisation possible de cet intervalle est le suivant. on désire par exemple, piloter une pression à quelque part sur la structure et on s'attend à garder cette pression positive. En fixant ϵ_{tarmin} à 0, cela permet de ne conserver que les valeurs de pilotage positives, si on trouve au moins une valeur de pilotage positive lors de la résolution du pilotage.

3.11.7 Opérande ETA_PILO_MAX / ETA_PILO_MIN

◇ ETA PILO MAX : etamax

Arrêt du calcul lorsque le paramètre de pilotage atteint la valeur donnée et_{amax} .

◇ ETA PILO MIN : etamin

Permet d'interrompre le calcul lorsque le paramètre `ETA_PILOTAGE` atteint cette valeur minimale `etamin` (pour des modèles adoucissants, permet de stopper le calcul lorsque la structure est suffisamment adoucie).

Attention :

Avec la loi ENDO ISOT BETON, ces deux mots clés sont obligatoires.

3.11.8 Opérande 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 $\text{eta} > \text{etamax} \rightarrow \text{eta} = \text{etamax}$; si $\text{eta} < \text{etamin} \rightarrow \text{eta} = \text{etamin}$), ce qui permet, en cas de convergence d'arrêter le calcul précisément sur etamin ou etamax.

Avec PROJ_BORNE='NON', on ne fait rien, donc le calcul s'arrêtera, en cas de convergence, avec une valeur supérieure à etamax ou inférieure à etamin.

3.11.9 Opérande SELECTION

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

Cet opérande 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.

3.12 Mot clé SOLVEUR

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

3.13 Mot clé CONVERGENCE

◇ CONVERGENCE :

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

L'algorithme continue les itérations globales tant que :

$$\max_{i=1, \dots, nb_ddl} |F_i^n| > \text{resrel} \max |\mathbf{L}|$$

où 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**

L'algorithme continue les itérations globales tant que :

$$\max_{i=1,\dots,nb_ddl} |F_i^n| > \text{resmax}$$

où F^n est le résidu de l'itération n (Cf. [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.

3.13.2 Opérande **RESI_REFE_RELA**

	RESI_REFE_RELA	=	resref ,	[R]
◇	SIGM_REFE	=	sigref ,	[R]
◇	EPSI_REFE	=	epsref ,	[R]
◇	FLUX_THER_REFE	=	fthref ,	[R]
◇	FLUX_HYD1_REFE	=	fh1ref ,	[R]
◇	FLUX_HYD2_REFE	=	fh2ref ,	[R]

Cet opérande conduit à estimer la convergence de l'algorithme de Newton de la manière suivante (Cf. R5.03.01 pour plus de détails). A partir de la contrainte de référence **sigref** (et/ou une déformation de référence **epsref** si l'on utilise des lois non locales à gradient de déformation, et/ou un flux thermique de référence **fthref** dans un cas THM, et/ou deux références de flux hydriques **fh1ref** et **fh2ref** dans un cas HHM), 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 :

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

3.13.3 Opérande **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.

3.13.4 Opérande **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 [§3.8.2]). 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**.

3.13.5 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).

/ 'NON'

Si maglob est insuffisant pour vérifier les critères de convergence donnés par l'utilisateur, on passe quand même à l'instant suivant. Utilisation à éviter.

3.13.6 Opérandes RESI_INTE_RELA / ITER_INTE_MAXI

```
◇ RESI_INTE_RELA = / 1.E-6 [DEFAULT]
                  / resint
◇ ITER_INTE_MAXI = / 10 [DEFAULT]
                  / iteint
```

Dans la plupart des relations de comportement, une équation non linéaire ou un système non linéaire doivent être résolus localement (en chaque point de GAUSS). Ces opérandes (résidu et nombre maximum d'itérations dites internes) sont utilisés pour tester la convergence de cet algorithme itératif de résolution. Pour plus de détails, se reporter à la documentation de référence, par exemple au document [R5.03.02]. Ces opérandes sont **inutiles** avec les comportements ELAS, VMIS_CINE_LINE, VMIS_ECMI_LINE, VMIS_ECMI_TRAC, VMIS_ISOT_LINE, VMIS_ISOT_TRAC, VISC_ISOT_LINE, VISC_ISOT_TRAC, BARENBLATT, NORTON_HOFF, DIS_CONTACT, DIS_CHOC, ARME, ASSE_CORN, DIS_GOUJ2E_PLAS, DIS_GOUJ2E_ELAS, VMIS_ASYM_LINE, GRILLE_ISOT_LINE, GRILLE_CINE_LINE, GRILLE_PINTO_MEN, PINTO_MENEGOTTO, GRANGER_FP et GRANGER_FP_V (hors contrainte plane), BAZANT_FD et toutes les relations META_XXX.

3.13.7 Opérande ITER_INTE_PAS

```
◇ ITER_INTE_PAS = 0 [DEFAULT]
                  itepas
```

Permet de redécouper localement le pas de temps pour faciliter l'intégration de la relation de comportement aux points de GAUSS (pour les relations de CHABOCHE, VISC_TAHERI, LMARC, LAIGLE, MONOCRISTAL, ROUSS_PR, ROUSS_VISC, CJS et BETON_DOUBLE_DP). Si itepas vaut 0, 1 ou -1 il n'y a pas de redécoupage. Si itepas est positif, on redécoupe systématiquement le pas de temps localement en itepas petits pas de temps avant d'effectuer l'intégration de la relation de comportement. Si itepas est négatif, le redécoupage en |itepas| petits pas de temps n'est effectué qu'en cas de non convergence locale.

3.13.8 Opérande RESO_INTE

```
◇ RESO_INTE = / 'IMPLICITE' [DEFAULT]
              / 'RUNGE_KUTTA_2'
              / 'RUNGE_KUTTA_4'
```

Permet de préciser le type de schéma d'intégration pour résoudre le système d'équations non linéaires formé par les équations constitutives des modèles de comportement à variables internes :

- les modèles POLY_CFC et POLYCRISTAL sont traités uniquement par le schéma explicite de RUNGE-KUTTA d'ordre 2,
- les deux modèles VMIS_POU_LINE et VMIS_POU_FLEJOU peuvent être traités par les deux schémas IMPLICITE et RUNGE_KUTTA_4,
- les deux modèles MONOCRISTAL et VENDUCHAB peuvent être traités par les deux schémas IMPLICITE et RUNGE_KUTTA_2,
- les autres modèles utilisent le schéma IMPLICITE.

3.14 Mot-clé CRIT_FLAMB

```

◇ CRIT_FLAMB = F (
    ◇ NB_FREQ =      / 3,          [DEFAULT]
                    / nbfreq,     [I]
    ◇ CHAR_CRIT =    / (-10,10),  [DEFAULT]
                    / intcc,
),

```

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 $\det(K^T - \lambda K^g) = 0$. K^T est la matrice tangente cohérente à cet instant. 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 (cd *MODE_ITER_SIMULT*). Ceci peut être assez coûteux pour les problèmes de grande taille.

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.

Pour les grands déplacements et les grandes déformations *GREEN(_GR)* ou *SIMO_MIEHE*, on résout $\det(K^T - \lambda Id) = 0$ car K^T contient alors K^g (et éventuellement K^p).

Le critère est alors un critère d'instabilité : quand λ change de signe (donc passe par 0) le chargement est instable.

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.

On stocke le mode propre correspondant à la plus petite charge critique (en valeur absolue) dans la *S.D. RESULTAT*, 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.

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 =

```

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.

Remarque :

| En présence de contact, on ne peut pas archiver plus de 99 999 instants de calculs

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*) à condition que l'évolution soit ordonnée (*EVOLUTION* : *CHRONOLOGIQUE* ou *RETROGADE*, cf [§3.6.1]) 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.

Deux 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

3.16.2 Opérande PRECISION

```

◇ PRECISION = prec
  Cf. [U4.71.00]

```

3.16.3 Opérande ARCH_ETAT_INIT / NUME_INIT / DETR_NUME_SUIV

```

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

```

Uniquement pour un *concept non réentrant* sinon message d'erreur. Permet d'imposer l'archivage de l'état initial dans le numéro d'ordre 0 (intéressant lorsque l'état initial provient d'un autre *STAT_NON_LINE*. Permet d'avoir le 1^{er} point sur une courbe).

```

/ 'NUME_INIT'      = nuinit

```

Uniquement pour un *concept réentrant* sinon message d'erreur. Permet de préciser à partir de quel numéro d'ordre on archive.

Par défaut :

- si l'état initial n'est pas fixé par le concept calculé, il s'agit du dernier numéro d'ordre +1 (exemple A),
- si le concept calculé coïncide avec le concept qui fixe l'état initial, il s'agit du numéro d'ordre +1 sous *ETAT_INIT* (exemple B et C).

```

◇ 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.

A - Exemple simple

```

LIST =      DEFI_LIST_REEL(  DEBUT =0.,
                             INTERVALLE =_F(JUSQU'A =5., NOMBRE =5)),

U1 = STAT_NON_LINE( INCREMENT =_F(      LIST_INST =LIST,
                                       INST_FIN =3.)) ,

U2 = STAT_NON_LINE( INCREMENT =_F(LIST_INST =LIST)) ,

U2 = STAT_NON_LINE( reuse=U2,
                    ETAT_INIT =_F(EVOL_NOLI =U1),
                    INCREMENT =_F(LIST_INST =LIST),
                    ARCHIVAGE =_F(LIST_INST = LIST)) ,

```

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

numéro d'archivage	: 1	2	3	4	5	6	7
instants correspondants	: 1.	2.	3.	4.	5.	4.	5.

B - Exemple simple

```
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 1er U2 est le suivant :

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

Le résultat final de l'archivage pour U2 est le suivant (par défaut nuinit = 3):

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

C - Exemple avec NUME_INIT

```
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,
                                                          NUME_INIT =2 ,
                                                          DETR_NUME_SUIV ='OUI')) ,
```

Le résultat de l'archivage pour le 1er U2 est le suivant :

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

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

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

3.16.4 Opérande CHAM_EXCLU

```
◇ CHAM_EXCLU = | 'DEPL'
                  | 'SIEF_ELGA'
                  | 'VARI_ELGA'
                  | 'VARI_NON_LOCAL'
                  | 'LANL_ELGA'
```

Permet de préciser les champs qui ne seront pas archivés, excepté au dernier pas de temps.

3.17 Opérande OBSERVATION

La syntaxe de ce mot clé commun à la commande *DYNA_NON_LINE* est décrite dans le document [U4.53.01].

3.18 Opérande SOLV_NON_LOCAL

La syntaxe de ce mot clé est identique au mot clé *SOLVEUR* décrit dans le document [U4.50.01]. A utiliser pour un modèle non local.

3.19 Opérande LAGR_NON_LOCAL

L'intégration de lois de comportement non locales impose la résolution d'un problème global (sur toute la structure) : la minimisation d'une fonctionnelle énergie (l'expression du lagrangien augmenté) par rapport à une variable nodale scalaire.

La résolution de ce problème s'effectue au moyen d'un algorithme newton primal et BFGS dual combiné, qui consiste en deux phases :

- Résolution du problème primal :
 - Minimisation par rapport à la variable interne non locale et son gradient (*cham_elem*)
 - Minimisation par rapport à la variable interne aux nœuds (*cham_no*)
 - Test de convergence primal : la plus grande composante du résidu assemblé
- Résolution du problème dual : (Maximisation par rapport aux multiplicateurs de Lagrange)
 - Calcul d'une direction de descente BFGS
 - Recherche linéaire par méthode de Wolfe
 - Test de convergence dual : la plus grande composante du gradient
 - Réactualisation des multiplicateurs de Lagrange

◇ **ITER_PRIM_MAXI : iterprimmax (10 par défaut)**

Nombre d'itérations maximales pour la résolution du problème primal.

◆ **RESI_PRIM_ABSO : resiprimab**

Précision pour le test de convergence pour le problème primal.

◇ **ITER_DUAL_MAXI : iterdmax (50 par défaut)**

Nombre d'itérations maximales pour la résolution du problème dual.

◆ **RESI_DUAL_ABSO : residabso**

Précision pour le test de convergence pour le problème dual.

◇ **R : rho (1000 par défaut)**

Coefficient de pénalisation du lagrangien augmenté.

Remarque :

Comme la précision du problème dual dépend fortement de celle du problème primal, on conseille de choisir une meilleure précision pour le problème primal, par exemple 100 ou 1000 fois plus que pour le problème dual.

3.20 Opérande INFO

◇ INFO : inf

Permet d'effectuer dans le fichier message diverses impressions intermédiaires en présence de contact unilatéral traité par la méthode des contraintes actives.

inf	= 1	impression de la liste des nœuds en contact après convergence à chaque itération de Newton.
	= 2	idem 1 plus impression des associations/dissociations de nœuds entre itérations de la méthode des contraintes actives.

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.

3.21 Opérande TITRE

◇ TITRE : tx

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