Opérateur THER_NON_LINE

1 But

Calculer la réponse thermique avec des non linéarités de comportements et de conditions aux limites.

L'équation de la chaleur est résolue en régime évolutif (sauf si aucune liste d'instant n'est fournie, seul le régime stationnaire est alors calculé). Les non linéarités proviennent soit du comportement (caractéristiques du matériau dépendant de la température), soit des conditions aux limites (rayonnement en milieu infini, flux non linéaire, source thermique dépendant de la température). Une formulation en enthalpie a été choisie afin de prendre en compte plus facilement les changements de phase du matériau.

Le calcul évolutif peut être initialisé, au premier instant de trois manières différentes (mot clé TEMP_INIT) :

- par une température constante,
- par un champ de température, défini au préalable, ou extrait d'un calcul précédent,
- par un calcul stationnaire.

Cet opérateur permet aussi de résoudre les problèmes de séchage (non linéaire) en résolvant l'équation de la chaleur où la concentration en eau C est assimilée à une température, pour la résolution. La conductivité thermique est dans ce cas le coefficient de diffusion, non linéaire en C et fonction, éventuellement, d'une température calculée au préalable.

Pour modéliser l'hydratation du béton, l'opérateur permet également d'ajouter un terme source fonction de la variable d'hydratation à l'équation de la chaleur. Ce terme est alors donné par une équation d'évolution où la température intervient.

Le concept produit par l'opérateur THER_NON_LINE est de type evol_ther comme pour une analyse linéaire par THER_LINEAIRE [U4.54.01].

Quand un calcul de sensibilité du résultat par rapport à un paramètre est demandé, il y a production d'autant de structures de données de type evol_ther que de paramètres requis.

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

2 Syntaxe

```
temper [evol_ther] = THER_NON_LINE

    (  ◊  RESULTAT  =      temper,

       ♦  MODELE       =  mo,                                  [modele]

       ♦  CHAM_MATER   =  chmat,                               [cham_mater]

       ◊  CARA_ELEM    =  carac,                               [cara_elem]

       ♦  EXCIT  = _F(
                      ♦  CHARGE    =  char,                   [charge]
                      ◊  FONC_MULT =  fonc,                   [fonction]
                    ),

       ◊  ETAT_INIT = _F(
                      ◊  /  STATIONNAIRE =  'OUI',            [DEFAUT]
                         /  VALE         =  tinit,            [R]
                         /  CHAM_NO      =  cinit,            [cham_no]
                         /  EVOL_THER    =  temp,             [evol_ther]
                            /  NUME_ORDRE   =  nuini_evol,    [I]
                            /  INST         =  inst_evol,     [R]
                               /  CRITERE   =  'RELATIF',     [DEFAUT]
                                  ◊  PRECISION =  / 1.E-6,    [DEFAUT]
                                                  / prec,     [R]
                               /  CRITERE   =  'ABSOLU',
                                  ♦  PRECISION =  prec,       [R]
                      ◊  INST_ETAT_INIT  =  inst_init,        [R]
                      ),

       ◊  INCREMENT    =_F(
             ... voir STAT_NON_LINE [U4.51.03] mot-clé INCREMENT ...
                          ),
◊  COMPORTEMENT  = _F(
        ♦ RELATION =   /'THER_NL',                            [DEFAUT]
                       /'THER_HYDR',
                       /'SECH_GRANGER',
                       /'SECH_MENSI',
                    /'SECH_BAZANT',
                    /'SECH_NAPPE',
       ◊  /  TOUT      =  'OUI',
          /  GROUP_MA  =  l_grmail,                           [l_gr_ma]
                        ),

◊  EVOL_THER_SECH  =         resuther,                        [evol_ther]
```


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

```
          ◊  METHODE          =  /'NEWTON',                          [DEFAUT]
                                 /'NEWTON_KRYLOV',
                                 / 'MODELE_REDUIT',
          ◊  MODELE_REDUIT    =  _F(
               REAC_ITER          =  / 0 ,
[DEFAUT]
                                    /it,                             [I]
             ♦  BASE_PRIMAL     =  mode_empi
             ◊  DOMAINE_REDUIT  =  /'NON'                            [DEFAUT]
                                    /'OUI'
      Si DOMAINE_REDUIT = 'OUI'
            ♦  GROUP_NO_INTERF = grno
                           ),
          ◊ NEWTON  =_F(
                              ◊  REAC_ITER =      /0,           [DEFAUT]
                                                  /it,          [I]
                              ◊  RESI_LINE_RELA=  /1.E-3,       [DEFAUT]
                                                  /reslin,      [R]
                              ◊  ITER_LINE_MAXI=  /0,           [DEFAUT]
                                                  /iterl,       [R]
                            ),

          ◊ CONVERGENCE=_F (
                              ◊  RESI_GLOB_RELA=  /1.E-6,       [DEFAUT]
                                                  /testr,       [R]
                              ◊  RESI_GLOB_MAXI=  /testl,       [R]
                              ◊  ITER_GLOB_MAXI=  /10,          [DEFAUT]
                                                  /iterl,       [R]
                            ),

          ◊  PARM_THETA =        /  theta,                       [R]
                                 /  0.57,                        [DEFAUT]

          ◊  SOLVEUR   = _F(
                              ... voir [U4.50.01] ...
                           ),

          ◊  ARCHIVAGE = _F(
              ... voir STAT_NON_LINE [U4.51.03] mot-clé ARCHIVAGE ...
                           ),

          ◊  OBSERVATION  = _F(
              ... voir STAT_NON_LINE [U4.51.03] mot-clé OBSERVATION ...
                           ),

          ◊  AFFICHAGE = _F(
              ... voir STAT_NON_LINE [U4.51.03] mot-clé AFFICHAGE ...
                           ),

          ◊  TITRE =             titre,                           [l_Kn]


        )
```


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

3 Opérandes

3.1 Opérande RESULTAT

♦ RESULTAT

Nom de l'objet résultat à enrichir en cas de poursuite du calcul (voir aussi ETAT_INIT).

3.2 Opérande MODELE

♦ MODELE = mo

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

3.3 Opérande CHAM_MATER

♦ CHAM_MATER = chmat

Nom du champ de matériau affecté sur le modèle.

3.4 Opérande CARA_ELEM

◊ CARA_ELEM = carac

Ce mot-clé est seulement nécessaire dans le cas d'une résolution de thermique non-linéaire avec matériau orthotrope (THER_NL_ORTH). Il permet alors de transmettre à l'opérateur de résolution la définition des axes locaux du repère d'orthotropie dans le milieu 2D ou 3D (commande AFFE_CARA_ELEM, mot clé MASSIF).

3.5 Mot clé EXCIT

♦ EXCIT =

Mot clé facteur permettant de définir plusieurs chargements. Pour chaque occurrence du mot clé facteur, on définit une charge éventuellement multipliée par une fonction du temps.

3.5.1 Opérande CHARGE

♦ CHARGE = char

Concept de type charge produit par AFFE_CHAR_THER ou par AFFE_CHAR_THER_F [U4.44.02] et par AFFE_CHAR_CINE ou par AFFE_CHAR_CINE_F [U4.44.03].

Remarque importante :

Pour chaque occurrence du mot clé facteur EXCIT les différents concepts char utilisés doivent être construits sur le même modèle mo .

3.5.2 Opérande FONC_MULT

◊ FONC_MULT = fonc

Coefficient multiplicatif fonction du temps (concept de type fonction, nappe ou formule) appliqué à la charge.

Remarque importante :

L'utilisation concomitante de FONC_MULT avec une charge contenant des chargements thermiques dépendant de la température est interdite ; c'est-à-dire pour des chargements de type ECHANGE_ , RAYONNEMENT, SOUR_NL ou FLUNL .

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

3.6 Mot clé ETAT_INIT

◊ ETAT_INIT =

Permet de définir le champ initial à partir duquel le calcul évolutif est effectué. Le champ initial est affecté du numéro d'ordre 0 et l'instant initial prend comme valeur le premier réel de la liste d'instant telle que définie par INCREMENT.

Remarque :

Si le mot clé ETAT_INIT est absent, on effectue uniquement le calcul stationnaire au premier instant défini sous le mot clé INCREMENT .

3.6.1 Opérande STATIONNAIRE

/ STATIONNAIRE = 'OUI'

La valeur initiale est alors le résultat d'un calcul stationnaire préalable. Ce calcul prend en compte les conditions aux limites définies sous le mot clé CHARGE et les caractéristiques matériaux à une température donnée. Par défaut, cette température est zéro. Néanmoins, il est possible de prendre un autre champ de température initiale en utilisant VALE, CHAM_NO ou EVOL_THER.

3.6.2 Opérande VALE

/ VALE = tinit
La valeur initiale de température est prise constante sur toute la structure.

3.6.3 Opérande CHAM_NO

/ CHAM_NO = cinit
La valeur initiale est définie par un cham_no de température (résultat de l'opérateurs CREA_CHAMP [U4.72.04]).

3.6.4 Opérande EVOL_THER

/ EVOL_THER = temp
La valeur initiale est extraite d'une structure de données de type evol_ther.

3.6.5 Opérande NUME_ORDRE/INST

◊ /NUME_ORDRE = nuini_evol
/INST = instini_evol

Numéro d'ordre du champ à extraire de cette structure de donnée. Extraction de l'état thermique initial dans l'evol_ther_temp 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.

Remarque :

Attention, il s'agit du numéro d'ordre dans la structure de donnée lue en reprise par le mot clé EVOL_THER précédent. Si cette structure de donnée a été calculée avec une liste d'instants différente de celle utilisée sous le mot clé facteur INCREMENT de la résolution courante, il est impératif de renseigner NUME_ORDRE sous INCREMENT , la même valeur de numéro d'ordre correspondant à des instants physiques différents. Dans le cas où les deux listes d'instants sont identiques, on peut se dispenser de renseigner deux fois le même NUME_ORDRE , sous ETAT_INIT et sous INCREMENT .

3.6.6 Opérande INST_ETAT_INIT

◊ INST_ETAT_INIT = istetaini

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

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_evol).

3.6.7 Opérande PRECISION/CRITERE

Confer [U4.71.00].

3.7 Mot clé INCREMENT

♦ INCREMENT =_F

Définit la liste des instants de calcul. Les opérandes du mot clé INCREMENT ont la même signification que dans le document [U4.51.03].

Remarque :

la liste d'instants contient obligatoirement au moins deux instants sinon un message d'erreur apparaît. Néanmoins cette obligation disparaît lorsqu'on fait un calcul stationnaire (mot-clef STATIONNAIRE dans ETAT_INIT). Dans ce cas le résultat de calcul ne contiendra que l'instant initial, calculé avec l'hypothèse de stationnarité.

3.8 Mot-clé COMPORTEMENT

◊ COMPORTEMENT =

La résolution du séchage a été ajoutée dans *Code_Aster* du fait de l'analogie des équations de la thermique et du séchage. Cela suppose d'assimiler la variable de calcul du séchage, la concentration en eau, à une variable de type 'TEMP' pendant la résolution.

Par défaut, la résolution effectuée sera de la thermique non linéaire. Ce mot clé facteur permet donc de distinguer la résolution de séchage de la thermique. De plus, l'équation du séchage est caractérisée par un coefficient de diffusion qui peut s'exprimer sous diverses formes. Ce mot clé facteur permet aussi de choisir l'une des équations du séchage, défini par l'expression de son coefficient de diffusion, disponible dans Aster. Pour effectuer un calcul de thermique non linéaire, ce mot clé devient optionnel, et la notion de comportement est transparente pour l'utilisateur.

Remarque :

Si le mot clé COMPORTEMENT est absent, on effectue un calcul de thermique non linéaire "standard".

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

3.8.1 Opérande RELATION

♦ RELATION: / 'THER_NL' [DEFAUT]
/ 'THER_HYDR'
/ 'SECH_GRANGER'
/ 'SECH_MENSI'
/ 'SECH_BAZANT'
/ 'SECH_NAPPE'

La syntaxe et le traitement de ce mot clé sont analogues à l'utilisation du mot clé de même nom de l'opérateur STAT_NON_LINE.

/ 'THER_NL'

Résolution thermique non linéaire standard.

Modélisations supportées :
- milieux continus 3D : 3D
- milieux continus 2D : 2D, AXIS

/ 'THER_HYDR'

Résolution de l'équation de la chaleur avec un terme source supplémentaire : $Q\dot{\xi}$

Q est la chaleur d'hydratation, supposée constante. La variable d'hydratation ξ est solution de la loi d'évolution non linéaire, résolue simultanément au problème de thermique :

$$\dot{\xi}=A(\xi)e^{-E/RT}$$

On se reportera à la documentation de l'opérateur DEFI_MATERIAU pour la signification des différents paramètres.

A l'instant initial du calcul, la variable d'hydratation prend pour valeur le champ trouvé sous le nom 'HYDR_ELNO' dans la structure de données evol_ther définie sous ETAT_INIT. Par défaut, l'état hydrique initial est pris vierge $\xi=0$.

Modélisations supportées :
- milieux continus 3D : 3D
- milieux continus 2D : 2D, AXIS

/ 'SECH_GRANGER'

Résolution du séchage caractérisé par l'équation $\frac{\partial C}{\partial t}-\mathrm{Div}\left[D(C,T)\nabla C\right]=0$

C'est l'équation de la chaleur non linéaire où la variable de séchage C tient le rôle de la température. Le choix de la relation de comportement permet de définir le coefficient de diffusion $D(C,T)$ selon diverses formes usuelles de la littérature. La formulation de Granger du coefficient de diffusion est donnée par l'expression :

$$D(C,T)=A\exp(BC)\frac{T}{T_0}.\exp\left(\frac{Q_s}{R}\right)\left(\frac{1}{T}-\frac{1}{T_0}\right)$$

On se reportera à la documentation de l'opérateur DEFI_MATERIAU pour la signification des différents paramètres. Dans le cas de l'utilisation de cette loi SECH_GRANGER, il est nécessaire de s'assurer de la cohérence entre le matériau utilisé et la loi de comportement : c'est à dire que le mot clé SECH_GRANGER a bien été renseigné dans DEFI_MATERIAU pour le matériau utilisé.

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

Modélisations supportées :
- milieux continus 3D : 3D
- milieux continus 2D : 2D, AXIS

Comme la résolution du séchage est effectuée par un opérateur de thermique, les modélisations supportées sont des modélisations thermiques, mais qui n'ont alors qu'une valeur conceptuelle d'ordre géométrique.

/ 'SECH_MENSI'

Résolution du séchage caractérisé par la loi de MENSI.
Dans le cas de l'utilisation de cette loi SECH_MENSI, il est nécessaire de s'assurer de la cohérence entre le matériau utilisé et la loi de comportement : c'est à dire que le mot clé SECH_MENSI a bien été renseigné dans DEFI_MATERIAU pour le matériau utilisé. Modélisations supportées : analogue à SECH_GRANGER.

/ 'SECH_BAZANT'

Résolution du séchage caractérisé par la loi de BAZANT.
Dans le cas de l'utilisation de cette loi SECH_BAZANT, il est nécessaire de s'assurer de la cohérence entre le matériau utilisé et la loi de comportement : c'est à dire que le mot clé SECH_BAZANT a bien été renseigné dans DEFI_MATERIAU pour le matériau utilisé. Modélisations supportées : analogue à SECH_GRANGER.

/ 'SECH_NAPPE'

Résolution du séchage avec un coefficient de diffusion défini par une nappe *Aster*.
Dans le cas de l'utilisation de cette loi SECH_NAPPE, il est nécessaire de s'assurer de la cohérence entre le matériau utilisé et la loi de comportement : c'est à dire que le mot clé SECH_NAPPE a bien été renseigné dans DEFI_MATERIAU pour le matériau utilisé. Modélisations supportées : analogue à SECH_GRANGER.

3.8.2 Opérandes TOUT / GROUP_MA

◊ / TOUT = 'OUI'
/ GROUP_MA = l_grmail

Spécifie les mailles sur lesquelles la relation de comportement est appliquée. De façon analogue à la mécanique, on peut utiliser plusieurs lois de séchage différentes, appliquées à des groupes de mailles complémentaires. Par contre, la thermique ne peut pas être mélangée à du séchage. Le comportement 'THER_NL' est nécessairement appliqué à tout le maillage, option TOUT : 'OUI', option par défaut, qui est en fait, dans le cas général, transparente pour l'utilisateur.

3.9 Opérande EVOL_THER_SECH

◊ EVOL_THER_SECH= resuther

Cet opérande est spécifique à la résolution du séchage. Le séchage est résolu après un calcul thermique préalable dans le cas général, (calcul non couplé mais chaîné thermique/séchage), le champ thermique intervenant comme variable auxiliaire, permettant de calculer le coefficient de diffusion de certaines lois. C'est une donnée d'entrée du calcul du séchage, qui doit être une structure de données de type evol_ther. Ce mot clé est obligatoire uniquement pour les lois 'SECH_GRANGER' et 'SECH_NAPPE', dont le coefficient de diffusion dépend de la température. La structure de données d'évolution thermique renseignée ici aura été obtenue par une exécution précédente d'un opérateur de thermique, linéaire ou non.

3.10 Opérande METHODE

◊ **METHODE = /'NEWTON'**
/ 'NEWTON_KRYLOV'
/'MODELE_REDUIT'

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

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]).

/'NEWTON_KRYLOV'

On utilise une version inexacte de l'algorithme de Newton-Raphson ; la précision des résolutions de systèmes linéaires par une méthode itérative est adaptée au cours de chaque pas de chargement (voir [R5.03.01]).

/'MODELE_REDUIT'

On utilise une méthode de réduction de modèle pour faire le calcul non-linéaire (voir [R5.01.05]). Il est nécessaire d'avoir construit une base réduire préalablement (commande DEFI_BASE_REDUITE).

3.11 Mot clé NEWTON

◊ NEWTON =

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

3.11.1 Opérande REAC_ITER

```
◊  REAC_ITER = /  0      [DEFAUT]
               /  it
```

La matrice utilisée pour les itérations globales de la méthode est la matrice tangente qui est réévaluée au début de chaque incrément de temps 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.

3.11.2 Opérande RESI_LINE_RELA / ITER_LINE_MAXI

```
◊  RESI_LINE_RELA  =  /  1.E-3     [DEFAUT]
                      /  reslin
◊  ITER_LINE_MAXI  =  /  0         [DEFAUT]
                      /  itlin
```

Ce sont les paramètres de la recherche linéaire qui permettent d'assurer une meilleure convergence de la méthode de NEWTON (Cf. [R5.03.01] pour plus de détails). On donne le nombre d'itérations maximum itelin à effectuer (la valeur par défaut 0 indique que l'on ne fait pas de recherche linéaire) et la précision reslin à atteindre pour réaliser la convergence de la recherche linéaire.

Remarque :

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.

3.12 Mot clé MODELE_REDUIT

Précise les caractéristiques de la méthode de résolution du problème incrémental non-linéaire par une méthode de réduction de modèle (voir [R5.01.05]).

```
◊  REAC_ITER       =  /0                  [DEFAUT]
```


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

```
                        /it
♦  BASE_PRIMAL      =   mode_empi
◊  DOMAINE_REDUIT   =   /'NON'              [DEFAUT]
                        /'OUI'
 Si DOMAINE_REDUIT = 'OUI'
       ♦  GROUP_NO_INTERF = grno
```

Les mots-clefs MATRICE, PREDICTION, REAC_ITER et REAC_INCR ont la même signification et le même usage que dans le mot-clef facteur NEWTON (voir §3.11).
Il est nécessaire de fournir une base empirique construite sur les déplacements (grâce à l'opérateur DEFI_BASE_REDUITE). Cette base doit avoir été construite sur le même modèle et le même maillage que le calcul THER_NON_LINE.
La réduction de modèle n'est pas compatible avec les fonctionnalités suivantes:

- Avec recherche linéaire
- Avec des conditions limites dualisées (AFFE_CHAR_THER, il faut utiliser AFFE_CHAR_CINE)

Il est possible d'activer l'hyper-réduction qui utilise une méthode de type DEIM. Dans ce cas, le calcul est réduit sur une zone du maillage restreinte (appelée RID) et construite à l'aide de l'opérateur DEFI_DOMAINE_REDUIT. Il faut donner le groupe de noeuds sur lequel est défini l'interface à l'aide du mot-clef GROUP_NO_INTERF.

3.13 Mot clé CONVERGENCE

◊ CONVERGENCE :

Permet de définir les valeurs associées aux critères de convergence.

Remarque :

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_RELA = / 1.e-6
                    / testr
```

L'algorithme continue les itérations externes tant que le résidu relatif est supérieur à testr.

$$\left(\sum_{i=1,\dots,nb\,ddl} (F_i^n)^2 \right)^{1/2} / \left(\sum_{i=1,\dots,nb\,ddl} (S_i)^2 \right)^{1/2} > testr$$

où F_i désigne le résidu et S_i le chargement thermique, l'indice n désigne la $n^{ième}$ itération.

3.13.2 Opérande RESI_GLOB_MAXI

```
◊  RESI_GLOB_MAXI   =   / 1.e-3
                        / testl
```

L'algorithme continue les itérations externes tant que le résidu absolu est supérieur à testl.

$$\max_{i=1,\dots,nb\,ddl} |F_i^n| > test1$$

où F_i désigne le résidu, l'indice n désigne la $n^{ième}$ itération.

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

3.13.3 Opérande ITER_GLOB_MAXI

◊ ITER_GLOB_MAXI = / 10
/ iterl

L'algorithme continue les itérations tant que leur nombre est inférieur à iterl.

3.14 Opérande PARM_THETA

◊ PARM_THETA = theta

L'argument theta est le paramètre de la théta-méthode appliquée au problème évolutif. Il doit être compris entre 0 (méthode explicite) et 1 (méthode totalement implicite). En l'absence, du mot clé, la valeur utilisée est $\theta=0.57$, un peu supérieure à $\theta=0.5$ correspond au schéma de Crank-Nicholson. L'incidence du choix de theta sur la stabilité de la méthode est détaillée dans [R5.02.02].

3.15 Mot clé SOLVEUR

◊ SOLVEUR =

Ce mot clé facteur est facultatif : il permet de choisir un autre solveur de résolution de système.

Cet opérande est commun à l'ensemble des commandes globales [U4.50.01].

3.16 Mot clé ARCHIVAGE

◊ ARCHIVAGE =

Ce mot clé est facultatif : par défaut, l'ensemble des champs calculés pour tous les pas calculés est archivé dans le concept resultat issu de la commande. Il sert à stocker certains numéros d'ordre dans une structure de données resultat et/ou exclure du stockage certains champs.

Ce mot-clé est identique à son équivalent pour l'opérateur STAT_NON_LINE, se référer à la documentation [U4.51.03] pour la description des sous mots-clés.

Remarque :

En cas d'arrêt du calcul par manque de temps CPU, les pas de temps précédemment calculés sont sauvegardés dans la base.

3.17 Mot clé OBSERVATION

◊ OBSERVATION =

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 [§3.13] (où on stocke tous les champs sur tout le modèle). Il sert essentiellement à des économies de stockage.

Ce mot clé est répétable et permet la création d'une table d'observation de même nom que le concept résultat de THER_NON_LINE.

Pour la description de la syntaxe de ce mot-clé facteur, se reporter à la documentation du mot-clé équivalent de DYNA_NON_LINE [U4.53.01].

3.18 Mot clé AFFICHAGE

◊ AFFICHAGE =

Ce mot-clef facteur permet de personnaliser l'affichage du tableau de convergence.

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

Si ce mot-clef n'est pas renseigné, le tableau est construit suivant les différentes options de calcul et avec INFO_RESIDU='NON'.

Pour la description de la syntaxe de ce mot-clé facteur, se reporter à la documentation du mot-clé équivalent de STAT_NON_LINE [U4.51.03].

3.19 Opérande TITRE

◊ TITRE = titre

Titre que l'on veut donner au résultat stocké dans temper, structure de données de type evol_ther [U4.03.01].

4 Modélisation

Les problèmes de thermique non linéaire peuvent être traités avec des modèles utilisant les éléments finis 3D, 2D ou AXIS décrits dans les documents [U3.22.01], [U3.23.01] et [U3.23.02] et [U3.24.01].

5 Exemple

On a défini ci-dessous les principales commandes utilisées pour effectuer un calcul de thermique non-linéaire transitoire. L'exemple indique comment poursuivre le calcul en enrichissant le concept résultat et comment préciser le champ "initial".

```
lr8    =  DEFI_LIST_REEL   ( ... )

conduc =  DEFI_FONCTION (      NOM_PARA  =  'TEMP',
                               VALE   = ...
                         )

enthal =  DEFI_FONCTION (      NOM_PARA  =  'TEMP',
                               VALE   = ... )

alu    =  DEFI_MATERIAU (THER_NL =_F(   LAMBDA =  conduc,
                                        BETA   =  enthal ) )
...
tempe     =  THER_NON_LINE (  MODELE=moth , CHAM_MATER=chmat,
                              ETAT_INIT    =_F(   VALE      =  20.0 ),
                              INCREMENT    =_F(   LIST_INST =  lr8,
                                                  NUME_FIN  = 20),
                              EXCIT        =_F(   CHARGE =  chth ),
                              CONVERGENCE  =_F(   RESI_GLOB_RELA   = 1.E-8,
                                                  ITER_GLOB_MAXI   = 20 ),
                           )
...
tempe  =  THER_NON_LINE    (  reuse  =  tempe,
                              MODELE=moth , CHAM_MATER=chmat,
                              ETAT_INIT    =_F(   EVOL_THER =  tempe,
                                                  NUME_ORDRE=  20 ),
                              INCREMENT    =_F(   LIST_INST =  lr8 ),
                              EXCIT        =_F(   CHARGE =  chth ),
                              CONVERGENCE  =_F(   RESI_GLOB_RELA   = 1.E-6,
                                                  ITER_GLOB_MAXI   = 10 ),
                           )
...
```


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