Manuel d'Utilisation
Fascicule U4.5- : Méthodes de résolution
Document : U4.54.02

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). 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.

2 Syntaxe

```
temper [evol_ther] = THER_NON_LINE

(  ◊  reuse  =      temper,

   ♦  MODELE       =      mo,                          [modele]

   ♦  CHAM_MATER   =      chmat,                       [cham_mater]

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

   ◊  TEMP_INIT =_F(
                  ◊  /  STATIONNAIRE =  'OUI',        [DEFAUT]
                     /  VALE         =  tinit,        [R]
                     /  CHAM_NO      =  tinit,        [cham_no_TEMP_R]
                     /  EVOL_THER    =  temp,         [evol_ther]
                        NUME_INIT    =  nuini_evol,   [I]
                  ),

   ◊  SENSIBILITE  =_F(
                     . . . voir [U4.50.02] . . .
                  ),

   ◊  INCREMENT =_F(
                     ♦  LIST_INST =     litps,        [listr8]
                     ◊  NUME_INIT =  /  0,
                                     /  nuini,        [I]
                     ◊  NUME_FIN  =     nufin,        [I]
                  ),

   ◊  COMP_THER_NL =_F(
                  ♦  RELATION =   / 'THER_NL',        [DEFAUT]
                                  / 'THER_HYDR',      [txm]
                                  / 'SECH_GRANGER',   [txm]
                                  / 'SECH_MENSI',     [txm]
                                  / 'SECH_BAZANT',    [txm]
                                  / 'SECH_NAPPE',     [txm]
                  ◊  /  TOUT   =         'OUI',       [txm]
                     /  |  GROUP_MA  =   l_grmail,    [l_gr_ma]
                        |  MAILLE    =   l_maille,    [l_ma]
                  ),

   ◊  EVOL_THER_SECH   =         resuther,              [evol_ther]
```

```
    ◊  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([U4.50.01])

    ◊  ARCHIVAGE =_F([U4.16.01])

    ◊  OPTION =             |  'FLUX_ELGA_TEMP',         [l_Kn]
                            |  'FLUX_ELNO_TEMP',

    ◊  TITRE =              titre,                        [l_Kn]

);
```

3 Opérandes

3.1 Opérande MODELE

♦ MODELE = mo

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

3.2 Opérande CHAM_MATER

♦ CHAM_MATER = chmat

Nom du champ de matériau affecté sur le modèle.

3.3 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.

3.3.1 Opérande CHARGE

♦ CHARGE = char

Concept de type charge produit par AFFE_CHAR_THER ou par AFFE_CHAR_THER_F [U4.44.02].

Remarque importante :

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

3.3.2 Opérande FONC_MULT

◊ FONC_MULT = fonc

Coefficient multiplicatif fonction du temps (concept de type fonction) 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 ou FLUNL .

3.4 Mot clé TEMP_INIT

◊ TEMP_INIT = litps

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 litps.

Remarque :

Si le mot clé TEMP_INIT est absent, on effectue uniquement le calcul stationnaire à l'instant défini sous le mot clé INCREMENT.

3.4.1 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.

3.4.2 Opérande VALE

/ VALE = tinit

La valeur initiale de température est prise constante sur toute la structure.

3.4.3 Opérande CHAM_NO

/ CHAM_NO = tinit

La valeur initiale est définie par un cham_no_TEMP_R (résultat des opérateurs AFFE_CHAM_NO [U4.44.11] ou RECU_CHAMP [U4.71.01]).

3.4.4 Opérande EVOL_THER

/ ♦ EVOL_THER = temp

La valeur initiale est extraite d'une structure de données de type evol_ther.

3.4.5 Opérande NUME_INIT

♦ NUME_INIT = nuini_evol

Numéro d'ordre du champ à extraire de cette structure de donnée.
Le champ initial est stocké dans la structure de données résultat temper sous le numéro d'ordre 0.

3.5 Mot clé SENSIBILITE

◊ SENSIBILITE = liste de paramètres sensibles

Active le calcul de la dérivée du champ de température par rapport à un paramètre sensible du problème.

Le document [U4.50.01] précise le fonctionnement du mot-clé.

3.6 Mot clé INCREMENT

◊ INCREMENT =

Permet de définir les instants de calcul qui déterminent les intervalles de temps pris pour intégrer l'équation différentielle.

Remarque :

Si le mot clé INCREMENT est absent, on crée une liste d'instants réduite au seul réel 0. et on effectue un calcul stationnaire.

3.6.1 Opérande LIST_INST

♦ LIST_INST = litps

Liste des instants produite par DEFI_LIST_REEL [U4.34.01].

3.6.2 Opérande NUME_INIT

◊ NUME_INIT = / 0
/ nuini

Indice de l'instant de calcul de départ dans la liste litps.

Si NUME_INIT est absent et si evol_ther est présent sous TEMP_INIT, alors nuini = nuini_evol.

3.6.3 Opérande NUME_FIN

◊ NUME_FIN = nufin

Indice de l'instant de calcul final dans la liste litps.

Les instants de calcul sont ceux définis dans le concept litps pris entre le nuini et le nufin numéro d'instant. Ainsi le premier pas de temps est défini entre l'instant correspondant à nuini et celui correspondant à nuini + 1. Le calcul, stationnaire, quand il est demandé, est fait à l'instant correspondant à nuini.

3.7 Mot clé COMP_THER_NL

◊ COMP_THER_NL =

La résolution du séchage a été ajoutée dans le *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 équation 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 COMP_THER_NL est absent, on effectue un calcul de thermique non linéaire "standard".

3.7.1 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 des mots clés COMP_INCR et COMP_ELAS 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.

Modélisations supportées :

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

/ 'SECH_GRANGER'

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

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 :

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

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 DEFI_MATERIAU pour le matériau utilisé.

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 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 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 DEFI_MATERIAU pour le matériau utilisé. Modélisations supportées : analogue à SECH_GRANGER.

3.7.2 Opérandes TOUT / GROUP_MA / MAILLE

◊ / TOUT = 'OUI'
/ GROUP_MA = l_grmail
/ MAILLE = l_maille

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.

3.8 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.

3.9 Mot clé NEWTON

◊ NEWTON =

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

3.9.1 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.

3.9.2 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.

3.10 Mot clé CONVERGENCE

◊ CONVERGENCE :

Permet de définir les valeurs associées aux critères de convergence :

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

3.10.1 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`.

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

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

3.10.2 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`.

$$\max_{i=1,\ldots,nb\ ddl} \left|F_i^n\right| > \texttt{testl}$$

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

3.10.3 Opérande ITER_GLOB_MAXI

◊ ITER_GLOB_MAXI = / 10
/ iterl

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

3.11 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].

3.12 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].

3.13 Mot clé ARCHIVAGE

◊ ARCHIVAGE =

Ce mot clé facteur est facultatif, par défaut l'ensemble des pas calculés est archivé dans le concept résultat issu de la commande.

Cet opérande est commun à l'ensemble des commandes globales [U4.16.01].

3.14 Opérande OPTION

◊ OPTION = 'FLUX_ELGA_TEMP'

Cette option effectue le calcul du flux de chaleur au points d'intégration à partir de la température.

◊ OPTION = 'FLUX_ELNO_TEMP'

Cette option effectue le calcul du flux de chaleur aux nœuds à partir de la température. Le calcul préalable de 'FLUX_ELGA_TEMP' n'est pas obligatoire.

3.15 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   (  DEBUT  =  0.D0,
                             INTERVALLE   =_F(   JUSQU_A=5.e-3, NOMBRE= 10 ),
                             INTERVALLE   =_F(   JUSQU_A=5.e-2, NOMBRE=  9 ),
                             INTERVALLE   =_F(   JUSQU_A=4.e-0, NOMBRE= 79 ),
                             INTERVALLE   =_F(   JUSQU_A=6.e-0, NOMBRE= 20 ),
                             )
conduc =  DEFI_FONCTION(
                                NOM_PARA  =  'TEMP',
                                VALE   =_F(     0.0 ,  210.0,
                                              660.0 ,  210.0,
                                              660.01,   95.0,
                                             1200.00,   95.0 )
                                PROL_DROIT      =  'LINEAIRE',
                                PROL_GAUCHE  =  'LINEAIRE',
                             )
enthal =  DEFI_FONCTION    (
                                NOM_PARA  =  'TEMP',
                                VALE   =_F(      0.0 , 0.0,
                                               660.0 , 1.980E9,
                                              660.01,  3.060E9,
                                             1200.00,  4.451E9 )
                                PROL_DROIT      =  'LINEAIRE',
                                PROL_GAUCHE  =  'LINEAIRE',
                             )
alu    =  DEFI_MATERIAU (THER_NL =_F(   LAMBDA =  conduc,
                                        BETA   =  enthal ) )
...
tempe     =  THER_NON_LINE (  MODELE=moth , CHAM_MATER=chmat,
                              TEMP_INIT    =_F(VALE     =  20.0 ),
                              INCREMENT    =_F(LIST_INST=  lr8),
                              EXCIT        =_F(CHARGE       =  chth ),
                              CONVERGENCE  =_F(RESI_GLOB_RELA     =1.E-6,
                                               ITER_GLOB_MAXI  =10 ),
                            )
...
tempe  =  THER_NON_LINE    (  reuse  =  tempe,
                             MODELE=moth , CHAM_MATER=chmat,
                              TEMP_INIT    =_F(EVOL_THER=  tempe,
                                               NUME_INIT =  20),
                              INCREMENT    =_F(LIST_INST=  lr8),
                              EXCIT        =_F(CHARGE       =  chth ),
                              CONVERGENCE  =_F(RESI_GLOB_RELA  =  1.E-6,
                                               ITER_GLOB_MAXI= 10 )
                             )
...
FIN();
```

Page laissée intentionnellement blanche.