
Opérateur THER_LINEAIRE

1 But

Résoudre un problème de thermique linéaire en régime stationnaire ou évolutif.

Le chargement thermique est défini par le mot clé `EXCIT`.

La discrétisation temporelle d'un calcul évolutif est fournie par la liste d'instants définie sous le mot clé `LIST_INST`. Ce calcul peut être initialisé, au premier instant, de trois manières différentes (mot clé `ETAT_INIT`) :

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

Le concept produit par cet opérateur est de type `evol_ther`.

2 Syntaxe

```
temper [evol_ther] = THER_LINEAIRE
( reuse = temper,
  ♦ MODELE = mo, [modele]
  ♦ CHAM_MATER = chmat, [cham_mater]
  ◇ CARA_ELEM = carac, [cara_elem]
  ♦ EXCIT = _F(
    ♦ CHARGE = char, [charge]
    FONC_MULT = fonc, [fonction, formule]
  ),
  ♦ ETAT_INIT = _F(
    / STATIONNAIRE = 'OUI', [DEFAULT]
    / VALE = tinit, [R]
    / CHAM_NO = tinit, [cham_no]
    / EVOL_THER = temp, [evol_ther]
    ◇ / NUME_ORDRE = nuini, [I]
    / INST = instini, [R]
    ◇ PRECISION = /1.0E-3, [DEFAULT]
    /prec, [R]
    ◇ CRITERE = /'RELATIF', [DEFAULT]
    /'ABSOLU',
  ),
  ◇ SENSIBILITE = _F(. . . voir [U4.50.02] . . . ),
  ◇ SENS_INIT = _F(
    / STATIONNAIRE = 'OUI', [DEFAULT]
    / EVOL_THER = temp, [evol_ther]
    NUME_INIT = nuini, [I]
  ),
  ◇ INCREMENT = _F(
    ♦ LIST_INST = litps, [listr8]
    ◇ / NUME_INST_INIT = nuini, [I]
    / INST_INIT = instini, [R]
    ◇ / NUME_INST_FIN = nufin, [I]
    / INST_FIN = instfin, [R]
    ◇ PRECISION = /1.0E-6, [DEFAULT]
  ),
  ◇ PARM_THETA = / theta, [R]
    / 0.57, [DEFAULT]
  ◇ SOLVEUR = _F(. . . voir [U4.50.01] . . . ),
  ◇ ARCHIVAGE = _F(. . . voir mot clé équivalent dans
    [U4.51.03] . . . ),
  ◇ 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 Opérande CARA_ELEM

◇ `CARA_ELEM = carac`

Le concept `carac` contient les caractéristiques des éléments de coque thermique, s'ils existent dans le modèle.

3.4 Mot clé EXCIT

♦ `EXCIT =`

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

3.4.1 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`.

3.4.2 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_`.*

3.5 Mot clé ETAT_INIT

◇ `ETAT_INIT =`

Permet de définir le champ initial à partir duquel le calcul évolutif est effectué.

Remarques :

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

Le champ initial est stocké dans la structure de données résultat `evol_ther` sous le numéro d'ordre 0.

3.5.1 Opérande STATIONNAIRE

```
/ STATIONNAIRE = 'OUI'
```

La valeur initiale du champ de température est alors le résultat d'un calcul stationnaire préalable.

3.5.2 Opérande VALE

```
/ VALE = tinit
```

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

3.5.3 Opérande CHAM_NO

```
/ CHAM_NO = tinit
```

La valeur initiale est définie par un `cham_no` de température (résultat de l'opérateur CREA_CHAMP [U4.72.04]).

3.5.4 Opérande EVOL_THER

```
/ EVOL_THER = temp
```

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

3.5.5 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'instant 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'instant sont identiques, on peut se dispenser de renseigner deux fois le même `NUME_ORDRE`, sous `ETAT_INIT` et sous `INCREMENT`.

3.5.6 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_evol`).

3.5.7 Opérande PRECISION/CRITERE

Cf. [U4.71.00].

3.6 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.

3.7 Mot clé SENS_INIT

◇ SENS_INIT =

Permet de définir le champ initial à partir duquel le calcul évolutif de la dérivée de la température est effectué, pour un calcul transitoire.

Remarque :

|Si le mot clé SENS_INIT est absent, l'initialisation est faite par un champ aux nœuds nul.

3.7.1 Opérande STATIONNAIRE

/ STATIONNAIRE = 'OUI'

La valeur initiale est celle d'un calcul stationnaire préalable. Cela n'est possible que si le même mode d'initialisation est retenu pour le calcul de la température.

3.7.2 Opérande EVOL_THER

/ EVOL_THER = temp

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

3.7.3 Opérande NUME_INIT

◇ NUME_INIT = nuini_evol

Numéro d'ordre du champ à extraire de cette structure de donnée désignée.

3.8 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.

3.8.1 Opérande LIST_INST

◆ LIST_INST = litps

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

3.8.2 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). 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_THER 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).

• En cas d'archivage (voir mot-clef ARCHIVAGE), l'instant initial en poursuite est le dernier pas archivé et non celui défini dans INST_INIT.

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

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

3.8.3 Opérande PRECISION/CRITERE

Cf. [U4.71.00].

3.9 Opérande PARM_THETA

```
◇ PARM_THETA =
```

L'argument `theta` est le paramètre de la thêta-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` correspondant au schéma de Crank-Nicholson. L'incidence du choix de `theta` sur la stabilité de la méthode est détaillée dans [R5.02.02].

3.10 Mot clé SOLVEUR

```
◇ SOLVEUR =
```

Ce mot clé facteur est facultatif : il permet de définir la méthode de résolution des systèmes linéaires. Cet opérande est commun à l'ensemble des commandes globales [U4.50.01].

3.11 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.

3.12 Opérande TITRE

```
◇ TITRE = titre
```

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

4 Modélisation

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

5 Exemple

5.1 Calcul transitoire

```
lr8 = DEFI_LIST_REEL ( DEBUT = 0.E0 ,  
                       INTERVALLE = (  
                           _F(JUSQU_A = 2.E-4 , NOMBRE = 2 ),  
                           _F(JUSQU_A = 1.E-3 , NOMBRE = 10 ),  
                           _F(JUSQU_A = 1.E-2 , NOMBRE = 9 ),  
                           _F(JUSQU_A = 1.E-1 , NOMBRE = 9 ),  
                           _F(JUSQU_A = 1.E+0 , NOMBRE = 9 ),  
                           _F(JUSQU_A = 2.0 , NOMBRE = 10 ),))  
  
tempe = THER_LINEAIRE ( MODELE = moth,  
                       CHAM_MATER = chmat,  
                       EXCIT = _F(CHARGE = chth),  
                       ETAT_INIT = _F(STATIONNAIRE = 'OUI'),  
                       INCREMENT = _F(LIST_INST = lr8,  
                                       NUME_INST_FIN = 30)  
                       )  
  
tempe = THER_LINEAIRE ( reuse = tempe,  
                       MODELE = moth,  
                       CHAM_MATER = chmat,  
                       EXCIT = _F(CHARGE = chth),  
                       ETAT_INIT = _F(EVOL_THER = tempe  
                                       NUME_ORDRE = 30),  
                       INCREMENT = _F(LIST_INST = lr8),  
                       )
```

Le premier appel à la commande THER_LINEAIRE permet d'effectuer un calcul stationnaire à l'instant 0. et d'enchaîner un calcul évolutif jusqu'à l'instant 0.1s (31 instants de calcul soit 30 calculs d'évolution).

Le second appel permet d'enrichir le concept tempe précédent, le calcul évolutif est poursuivi à partir du 31^{ème} instant de calcul.

5.2 Sensibilité à une température imposée

```
ta = DEFI_PARA_SENSI ( VALE = 70 )  
tb = DEFI_PARA_SENSI ( VALE = 30 )  
  
ca = AFFE_CHAR_THER_F ( MODELE = moth,  
                       TEMP_IMPO = ( _F(GROUP_MA = 'bord_sup',  
                                       TEMP = ta),  
                                       _F(GROUP_MA = 'bord_inf',  
                                       TEMP = tb) ) )  
  
tempe = THER_LINEAIRE ( MODELE = moth,  
                       CHAM_MATER = chmat,  
                       EXCIT = _F(CHARGE = chth),  
                       SENSIBILITE = ( ta , tb ),  
                       )
```

Ce calcul produira la structure de données tempe, de type evol_ther, contenant le champ de température sous le nom TEMP. Il produira aussi deux autres structures de type evol_ther. La première contiendra, sous le nom de champ TEMP, le champ de la dérivée de la température par rapport au paramètre ta. La seconde contiendra la dérivée par rapport au paramètre tb. Le nom de ces structures est créé automatiquement par le code et reste inconnu de l'utilisateur. L'accès à leur contenu (impression, test, post_releve ...) se fait en invoquant la commande correspondante avec le nom de la structure principale, temp, et le nom du paramètre sensible concerné, ta ou tb.

6 Remarque

La commande `CALC_ELEM` [U4.81.01] permet de calculer les flux de chaleur, aux points d'intégration ou aux nœuds, à partir du champ aux nœuds de température ainsi obtenu par `THER_LINEAIRE`.