

---

## Operator THER\_NON\_LINE\_MO

---

### 1 Goal

---

To calculate the nonlinear stationary thermal answer with mobile loading.

This operator allows to solve the equation of stationary heat in a mobile reference frame related to a loading and moving in a given direction and at a speed [R5.02.04].

Nonthe linearities of the problem come as well from the characteristics of material which depend on the temperature, as boundary conditions of type radiation.

The problems of this type can be treated with models using of the finite elements of structure plans, axisymmetric and three-dimensional.

The concept produced by this operator is of type `evol_ther` as for an analysis by `THER_LINEAIRE` [U4.54.01] or `THER_NON_LINE` [U4.54.02], but it contains one sequence number since a stationary calculation is carried out.

## 2 Syntax

```
temper [evol_ther] = THER_NON_LINE_MO

(
  ◆ MODEL          =          Mo,                [model]
  ◆ CHAM_MATER     =          chmat,            [cham_mater]
  ◆ EXCIT          =  _F (
    ◆ LOAD          =          tank,            [char_ther]
    ),
  ◇ ETAT_INIT      =  _F (
    ◆ EVOL_THER     =          temp,            [evol_ther]
    ◇ NUME_ORDRE    =  / nuini_evol,          [I]
    / 0,            [DEFECT]
    ),
  ◇ CONVERGENCE    =  _F (
    ◇ CRIT_TEMP_RELA =  / 1.E-3,            [DEFECT]
    / testr,        [R]
    ◇ CRIT_ENTH_RELA =  / 1.E-2,            [DEFECT]
    / test1,        [R]
    ◇ ITER_GLOB_MAXI =  / 10,              [DEFECT]
    / iter1,        [R]
    ),
  ◇ SOLVEUR        =  _F (
    ... to see document [U4.50.01]...
    ),
  ◇ TITLE          =          title,            [l_Kn]
)
)
```

## 3 Operands

### 3.1 Operand MODEL

- ◆ MODEL = Mo

Name of the model whose elements are the object of thermal calculation.

### 3.2 Operand CHAM\_MATER

- ◆ CHAM\_MATER = chmat

Name of the affected material field on the model.

### 3.3 Keyword EXCIT

- ◆ EXCIT =

Keyword factor allowing to define several loadings. For each occurrence of the keyword factor, one defines a time-constant load.

#### 3.3.1 Operand LOAD

- ◆ LOAD = tank

Concept of the type load product by AFFE\_CHAR\_THER or by AFFE\_CHAR\_THER\_F [U4.44.02].

#### Important remarks:

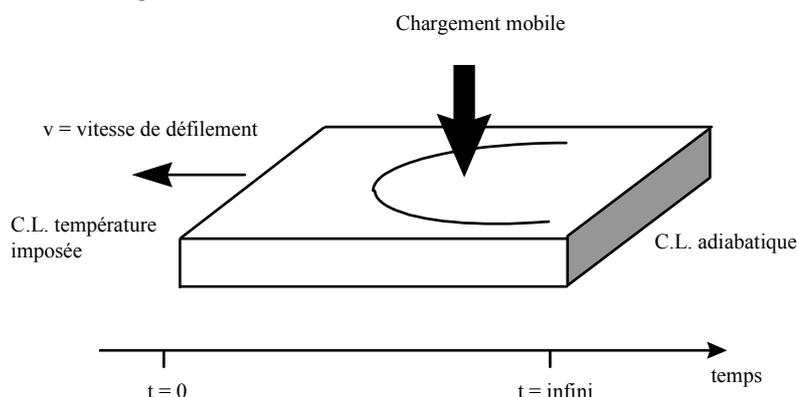
*For each occurrence of the keyword factor EXCIT various concepts tank used must be built on the same model Mo .*

*In addition, the operator AFFE\_CHAR\_THER or AFFE\_CHAR\_THER\_F must obligatorily inform under the keyword factor CONVECTION a concept of the type cham\_no\_DEPL\_R defining the rate of travel of the field and product by CREA\_CHAMP .*

*It is necessary to take care to define boundary conditions coherent with the run of the part (or of the loading, according to the selected reference frame) and the assumption of stationnarity:*

*If the solid is animated of a uniform translatory movement at constant speed, it is equivalent to consider the calculated field of temperature:*

- *like a photograph at a given moment,*
- *or like a temporal succession of the map of temperature in a section of fixed, orthogonal X-coordinate with the direction of run.*



*Thus, downstream in space (early in time), the boundary condition on the edge of the part must impose the temperature of the part before arrives the mobile loading. Upstream in space (or late in time), a heat flux no one (adiabaticity) guarantees the assumption of stationnarity.*

## 3.4 Keyword STATE\_INIT

◇ ETAT\_INIT

Allows to define the initial field from which calculation in pointer under condition of stationnarity is carried out.

### 3.4.1 Operand EVOL\_THER

◆ EVOL\_THER = temp

Name of the structure of data of the type `evol_ther` of which is extracted the field of temperature from which the solution with the problem is calculated.

### 3.4.2 Operand NUME\_ORDRE

◇ NUME\_ORDRE = / nuini\_evol  
/ 0 [DEFECT]

Sequence number of the field to be extracted from this structure of data.

#### Note:

*For this kind of calculation, there is no initial condition to define (it is defined with a boundary condition on an edge of the field defined by the model). But, one can nevertheless only carry out stationary calculation starting from a field of temperature already calculated as solution of the problem for part of the load. That can be useful if the application of the totality of the loading makes convergence difficult. One can then connect several resolutions with a loading which evolves gradually and take as reiterated first of a calculation running the solution of preceding calculation. Concretely, the keyword then is used STATE\_INIT to define the field of temperature with which one wants to initialize current calculation.*

## 3.5 Keyword CONVERGENCE

◇ CONVERGENCE =

Allows to define the values associated with the convergence criteria:

### 3.5.1 Operand CRIT\_TEMP\_RELA

◇ CRIT\_TEMP\_RELA = / 1.e-3  
/ testr

The algorithm continues the iterations as long as the relative variation of reiterated is higher than `testr`.

$$\left( \sum_{i=1, \dots, nbddl} (T_i^n - T_i^{n-1})^2 \right)^{1/2} / \left( \sum_{i=1, \dots, nbddl} (T_i^n)^2 \right)^{1/2} > testr$$

where  $T_i^n$  indicate the value of the temperature to node I for N<sup>ième</sup> iteration.

## 3.5.2 Operand CRIT\_ENTH\_REL

◇ CRIT\_ENTH\_REL = / 1.e-2  
/ test1

The algorithm continues the iterations as long as the relative variation of reiterated is higher than test1.

$$\left( \sum_{i=1, \dots, N} (\beta_i^n - \beta_i^{n-1})^2 \right)^{1/2} / \left( \sum_{i=1, \dots, N} (\beta_i^n)^2 \right)^{1/2} > test1$$

where  $\beta_i^n$  indicate the value of the enthalpy with N<sup>ième</sup> iteration with l<sup>ème</sup> not Gauss.

## 3.5.3 Operand ITER\_GLOB\_MAXI

◇ ITER\_GLOB\_MAXI = / 10  
/ iter1

The algorithm continues the external iterations as long as the iteration count external is lower than iter1.

## 3.6 Keyword SOLVEUR

◇ SOLVEUR =

This keyword factor is optional: it makes it possible to choose another solver of resolution of system.

This operand is common to the whole of the total orders [U4.50.01].

## 3.7 Operand TITLE

◇ TITLE = title

Title which one wants to give to the result stored in temp structure of data of the type evol\_ther [U4.03.01].

# 4 Modeling

The problems of nonlinear thermics in pointer and under condition of stationnarity can be treated with models using the finite elements 3D or 2D plan described in the documents [U3.23.01] and [U3.24.01].

## 5 Examples

One presents below the principal orders used to carry out a non-linear calculation of thermics in pointer and under condition of stationnarity.

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

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

chechmate = DEFI_MATERIAU (  THER_NL= _F (  LAMBDA = conduc,
                                           BETA  = enthal),)

...

chavi = CREA_CHAMP (  GRID   = e-mail, TYPE_CHAM=' NOEU_DEPL_R',
                    AFFE    = _F ( ALL= 'YES',
                    NOM_CMP = ('DX', 'DY'),
                    VALE_R  = (1.0, 0.0))

tank = AFFE_CHAR_THER (  MODEL      = model,
                       FLUX_REP   = _F ( GROUP_MA = penny,
                       FLUN       = 5.E6),
                       CONVECTION=_F ( SPEED = chavi))

temple = THER_NON_LINE_MO (  MODEL      = model,
                           CHAM_MATER = chmat,
                           EXCIT      = _F (LOAD = tank),
                           )

...
```