
Operator THER_NON_LINE

1 Goal

To calculate the thermal answer with nonlinearities of behaviors and boundary conditions.

The equation of heat is solved in evolutionary mode (except if no list of moment is provided, only the stationary mode is then calculated). Nonthe linearities come either from the behavior (characteristics of material depend on the temperature), or of the boundary conditions (radiation in infinite medium, nonlinear flow, thermal source depending on the temperature). A formulation in enthalpy was selected in order to more easily take into account the phase shifts of material.

Evolutionary calculation can be initialized, at the first moment in three different ways (keyword `TEMP_INIT`) :

- by a constant temperature,
- by a field of temperature, definite, or extracted as a preliminary from a preceding calculation,
- by a stationary calculation.

This operator also allows to solve the problems of drying (nonlinear) by solving the equation of heat where water concentration C is comparable at a temperature, for the resolution. Thermal conductivity is in this case the coefficient of diffusion, nonlinear in C and function, possibly, of a temperature calculated as a preliminary.

To model the hydration of the concrete, the operator also allows to add a term function source of the variable of hydration to the equation of heat. This term is then given by an equation of evolution where the temperature intervenes.

The concept produced by the operator `THER_NON_LINE` is of type `evol_ther` as for a linear analysis by `THER_LINEAIRE` [U4.54.01].

When a calculation of sensitivity of the result compared to a parameter is required, there is production of as many structures of data of the type `evol_ther` that necessary parameters.

2 Syntax

```

temper [evol_ther] = THER_NON_LINE

( ◇ RESULT      =  temper,                                [evol_ther]

  ◇ MODEL       =  Mo,                                    [model]

  ◇ CHAM_MATER  =  chmat,                                [cham_mater]

  ◇ EXCIT = _F(
      ◇ LOAD      =  tank,                                [load]
      ◇ FONC_MULT =  fonc,                                [function]
  ),

  ◇ ETAT_INIT = _F (
      ◇ / STATIONARY = 'YES',                            [DEFECT]
      / VALE        =  tinit,                            [R]
      / CHAM_NO     =  cinit,                            [cham_no]
      / EVOL_THER   =  temp,                             [evol_ther]
      / NUME_ORDRE  =  nuinievol,                       [I]
      / INST        =  instevol,                       [R]
      / CRITERION   =  'RELATIVE',                     [DEFECT]
      ◇ PRECISION   =  / 1.E-6,                       [DEFECT]
      / prec,                                             [R]
      / CRITERION   =  'ABSOLUTE',
      ◇ PRECISION   =  prec,                             [R]
      ◇ INST_ETAT_INIT = instinit,                     [R]
  ),

  ◇ INCREMENT    = _F (
      ... to see STAT_NON_LINE [U4.51.03] keyword INCREMENT...
  ),

  ◇ BEHAVIOR     = _F (
      ◇ RELATION  =  / 'THER_NL',                        [DEFECT]
      / 'THER_HYDR',
      / 'SECH_GRANGER',
      / 'SECH_MENSI',
      / 'SECH_BAZANT',
      / 'SECH_NAPPE',
      ◇ / ALL     =  'YES',
      / GROUP_MA  =  l_grmail,                          [l_gr_ma]
  ),

  ◇ EVOL_THER_SECH =  resuther,                          [evol_ther]

```

```

◇ METHOD = / 'NEWTON' , [DEFECT]
           / 'NEWTON_KRYLOV' ,
           / 'MODELE_REDUIT' ,
◇ MODELE_REDUIT = _F (
  REAC_ITER = / 0 ,
[DEFECT]
           /it, [I]
◇ BASE_PRIMAL = mode_empi
◇ DOMAINE_REDUIT = / 'NOT' [DEFECT]
                  / 'YES'
If DOMAINE_REDUIT = 'YES'
  ◇ GROUP_NO_INTERF = grno
  ),
◇ NEWTON = _F (
  ◇ REAC_ITER = /0, [DEFECT]
                /it, [I]
  ◇ RESI_LINE_RELA= /1.E-3, [DEFECT]
                    /reslin, [R]
  ◇ ITER_LINE_MAXI= /0, [DEFECT]
                    /iterl, [R]
  ),
◇ CONVERGENCE= _F (
  ◇ RESI_GLOB_RELA= /1.E-6, [DEFECT]
                    /testr, [R]
  ◇ RESI_GLOB_MAXI= /testl, [R]
  ◇ ITER_GLOB_MAXI= /10, [DEFECT]
                    /iterl, [R]
  ),
◇ PARM_THETA = / theta, [R]
               / 0.57, [DEFECT]
◇ SOLVEUR = _F (
  ... to see [U4.50.01]...
  ),
◇ FILING = _F (
  ... to see STAT_NON_LINE [U4.51.03] keyword FILING...
  ),
◇ OBSERVATION = _F (
  ... to see STAT_NON_LINE [U4.51.03] keyword OBSERVATION...
  ),
◇ TITLE = title, [l_Kn]
)

```

3 Operands

3.1 Operand RESULT

- ◆ RESULT

Name DE the result object to enrich in the event of continuation by calculation (see too ETAT_INIT).

3.2 Operand MODEL

- ◆ MODEL = Mo

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

3.3 Operand CHAM_MATER

- ◆ CHAM_MATER = chmat

Name of the affected material field on the model.

3.4 Keyword EXCIT

- ◆ EXCIT =

Keyword factor allowing to define several loadings. For each occurrence of the keyword factor, one defines a load possibly multiplied by a function of time.

3.4.1 Operand LOAD

- ◆ LOAD = tank

Concept of the type load product by AFFE_CHAR_THER or by AFFE_CHAR_THER_F [U4.44.02].

Notice important:

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

3.4.2 Operand FONC_MULT

- ◆ FONC_MULT = fonc

Multiplicative coefficient function of time (concept of the type function, tablecloth or formula) applied to the load.

Notice important:

The concomitant use of FONC_MULT with a load containing of the thermal loadings depending on the temperature is prohibited; i.e. for loadings of the type ECHANGE_ , RADIATION, SOUR_NL or FLUNL .

3.5 Keyword STATE_INIT

- ◆ ETAT_INIT =

Allows to define the initial field from which evolutionary calculation is carried out. The initial field is affected of the sequence number 0 and the initial moment takes as value the first reality of the list of moment as defined by INCREMENT.

Note:

Warning : The translation process used on this website is a "Machine Translation". It may be imprecise and inaccurate in whole or in part and is provided as a convenience.

Copyright 2019 EDF R&D - Licensed under the terms of the GNU FDL (<http://www.gnu.org/copyleft/fdl.html>)

If the keyword STATE_INIT is absent, one carries out only stationary calculation at the first moment defined under the keyword INCREMENT.

3.5.1 Operand STATIONARY

/ STATIONARY = 'YES'

The initial value is then the result of a preliminary stationary calculation. This calculation takes into account the boundary conditions defined under the keyword `LOAD` and characteristic materials at a given temperature. By default, this temperature is zero. Nevertheless, it is possible to take another field of initial temperature while using `VALE`, `CHAM_NO` or `EVOL_THER`.

3.5.2 Operand VALE

/ VALE = tinit

The initial value of temperature is taken constant on all the structure.

3.5.3 Operand CHAM_NO

/ CHAM_NO = cinit

The initial value is defined by one `cham_no` of temperature (result of the operators `CREA_CHAMP` [U4.72.04]).

3.5.4 Operand EVOL_THER

/ EVOL_THER = temp

The initial value is extracted from a structure of data of the type `evol_ther`.

3.5.5 Operand NUME_ORDRE/INST

◇ /NUME_ORDRE = nuini_evol
/INST = instini_evol

Sequence number of the field to be extracted from this structure of data. Extraction of the initial thermal state `invol_ther_temp` starting from the number of filing `NUME_ORDRE` or of the moment of filing `INST` to carry out the continuation of calculation. If `NUME_ORDRE` or `INST` are not filled, one takes the last existing number filed in `evol`.

Note:

Attention, it acts of the sequence number in the structure of data read in recovery by the keyword `EVOL_THER` precedent. If this structure of data were calculated with a list of moments different from that used under the keyword factor `INCREMENT` current resolution, it is imperative to inform `NUME_ORDRE` under `INCREMENT`, the same value of sequence number corresponding to different physical moments. If the two lists of moments are identical, one can exempt oneself to inform the same one twice `NUME_ORDRE`, under `ETAT_INIT` and under `INCREMENT`.

3.5.6 Operand INST_ETAT_INIT

◇ INST_ETAT_INIT = istetaini

One can associate a value of moment `istetaini` in this initial state. By default:

- when the initial state is defined by the data of the fields, an associated moment ago.
- when the state is given by a concept `evol_noli`, it is the moment in preceding calculation (`lsetetaini = instini_evol`).

3.5.7 Operand PRECISION/CRITERE

Confer [U4.71.00].

3.6 Keyword INCREMENT

Warning : The translation process used on this website is a "Machine Translation". It may be imprecise and inaccurate in whole or in part and is provided as a convenience.

Copyright 2019 EDF R&D - Licensed under the terms of the GNU FDL (<http://www.gnu.org/copyleft/fdl.html>)

◆ INCREMENT =_F

The list of the moments of calculation defines. Operands of the keyword INCREMENT have the same meaning as danS the document [U4.51.03].

Notice :

the list of moments contains obligatorily at least two moments if not an error message appears. Nevertheless this obligation disappears when one does a stationary calculation (keyword STATIONARY in ETAT_INIT). In this case the computation result will contain only the initial moment, calculated with the assumption of stationnarity.

3.7 Keyword BEHAVIOR

◇ BEHAVIOR =

The resolution of drying was added in Code_Aster because of analogy of the equations of thermics and drying. That supposes to assimilate the variable of calculation of drying, the water concentration, with a variable of the type 'TEMP' during the resolution.

By default, the resolution carried out will be nonlinear thermics. This keyword factor thus makes it possible to distinguish the resolution of drying from thermics. Moreover, the equation of drying is characterized by a coefficient of diffusion which can be expressed in various forms. This keyword factor also makes it possible to choose one of the equations of the drying, defined by the expression of its coefficient of diffusion, available in Aster. To carry out a nonlinear calculation of thermics, this keyword becomes optional, and the concept of behavior is transparent for the user.

Note:

If the keyword BEHAVIOR is absent, one carries out a calculation of nonlinear thermics "standard".

3.7.1 Operand RELATION

◆ RELATION: / `THER_NL' [DEFECT]
/ `THER_HYDR'
/ `SECH_GRANGER'
/ `SECH_MENSI'
/ `SECH_BAZANT'
/ `SECH_NAPPE'

The syntax and the treatment of this keyword are similar to the use of the same keyword name of the operator STAT_NON_LINE.

/ `THER_NL'

Standard nonlinear thermal resolution.

Supported modelings:

- continuous mediums 3D: 3D
- continuous mediums 2D: 2D, AXIS

/ `THER_HYDR'

Resolution of the equation of heat with an additional source term: $Q\dot{\xi}$

Q is the heat of hydration, presumedly constant. The variable of hydration ξ is solution of the nonlinear law of evolution, solved simultaneously with the problem of thermics:

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

One will refer to the documentation of the operator DEFI_MATERIAU for the significance of the various parameters.

At the initial moment of calculation, the variable of hydration takes for value the field found under the name `HYDR_ELNO' in the structure of data evol_ther defined under ETAT_INIT. By default, the initial hydrous state is taken virgin $\xi=0$.

Supported modelings:

- continuous mediums 3D: 3D
- continuous mediums 2D: 2D, AXIS

/ `SECH_GRANGER'

Resolution of the drying characterized by the equation $\frac{\partial C}{\partial t} - \text{Div}[D(C, T)\nabla C] = 0$

It is the equation of nonlinear heat where the variable of drying C the role of the temperature holds. The choice of the relation of behavior makes it possible to define the coefficient of diffusion $D(C, T)$ according to various usual forms of the literature. The formulation of Granger of the coefficient of diffusion is given by the expression:

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

One will refer to the documentation of the operator DEFI_MATERIAU for the significance of the various parameters. In the case of the use of this law SECH_GRANGER, it is necessary to make sure of coherence between material used and the law of behavior: i.e. that the keyword SECH_GRANGER indeed was well informed in DEFI_MATERIAU for material used.

Supported modelings:

- continuous mediums 3D: 3D
- continuous mediums 2D: 2D, AXIS

As the resolution of drying is carried out by an operator of thermics, supported modelings are thermal modelings, but which do not have whereas a conceptual value of a geometrical nature.

```
/ 'SECH_MENSI'
```

Resolution of the drying characterized by the law of MENSI.

In the case of the use of this law SECH_MENSI, it is necessary to make sure of coherence between material used and the law of behavior: i.e. that the keyword SECH_MENSI indeed was well informed in DEFI_MATERIAU for material used. Supported modelings: analogue with SECH_GRANGER.

```
/ 'SECH_BAZANT'
```

Resolution of the drying characterized by the law of BAZANT.

In the case of the use of this law SECH_BAZANT, it is necessary to make sure of coherence between material used and the law of behavior: i.e. that the keyword SECH_BAZANT indeed was well informed in DEFI_MATERIAU for material used. Supported modelings: analogue with SECH_GRANGER.

```
/ 'SECH_NAPPE'
```

Resolution of drying with a coefficient of diffusion defined by a tablecloth Aster.

In the case of the use of this law SECH_NAPPE, it is necessary to make sure of coherence between material used and the law of behavior: i.e. that the keyword SECH_NAPPE indeed was well informed in DEFI_MATERIAU for material used. Supported modelings: analogue with SECH_GRANGER.

3.7.2 Operands ALL/GROUP_MA

```
◇ / ALL = 'YES'  
/ GROUP_MA = l_grmail
```

Specify the meshes to which the relation of behavior is applied. In a way similar to mechanics, one can use several different laws of drying, applied to groups of complementary meshes. On the other hand, thermal cannot be to it mixed with drying. The behavior 'THER_NL' is necessarily applied to all the grid, option ALL: 'YES', option by default, which is in fact, in the case general, transparent for the user.

3.8 Operand EVOL_THER_SECH

```
◇ EVOL_THER_SECH= resuther
```

This operand is specific to the resolution of drying. Drying is solved after a preliminary thermal calculation in the case general, (calculation not coupled but chained thermal/drying), the thermal field intervening like auxiliary variable, making it possible to calculate the coefficient of diffusion of certain laws. It is an input datum of the calculation of the drying, which must be one structure of data of the type evol_ther. This keyword is obligatory only for the laws 'SECH_GRANGER' and 'SECH_NAPPE', of which the coefficient of diffusion depends on the temperature. The structure of data of here well informed thermal evolution will have been obtained by a preceding execution of an operator of thermics, linear or not.

3.9 Operand METHOD

```
◇ METHOD = / 'NEWTON'  
/ 'NEWTON_KRYLOV'  
/ 'MODELE_REDUIT'
```

Allows to choose the method of resolution of the nonlinear incremental problem.

`/ `NEWTON``

One uses the algorithm of Newton-Raphson to solve the problem (see [R5.03.01]).

`/ `NEWTON_KRYLOV``

One uses an inaccurate version of the algorithm of Newton-Raphson; the precision of the resolutions of systems linear by an iterative method is adapted during each step of loading (see [R5.03.01]).

`/ `MODEL_REDUCED``

One is used method of reduction of model to do the non-linear calculation (see [R5.01.05]). It is necessary to have built a base to reduce (order beforehand `DEFI_BASE_REDUITE`).

3.10 Keyword **NEWTON**

◇ `NEWTON =`

Specify the characteristics of the method of resolution of the nonlinear problem (method of NEWTON-RAPHSON).

3.10.1 Operand **REAC_ITER**

◇ `REAC_ITER = / 0 [DEFECT]
/ it`

The matrix used for the total iterations of the method is the tangent matrix which are revalued at the beginning of each increment of time and all them `it` iterations of NEWTON for an increment of time given (precisely to the iterations of number `it`, `2it`, `3it`...). Thus with the first iteration of NEWTON, one reassembles the tangent matrix only if `it 1` is worth: if not one keeps the matrix used in the phase of prediction. By convention if `it` the 0 matrix is worth is not revalued during all the step of time.

3.10.2 Operand **RESI_LINE_RELA/ITER_LINE_MAXI**

◇ `RESI_LINE_RELA = / 1.E-3 [DEFECT]
/ reslin`
◇ `ITER_LINE_MAXI = / 0 [DEFECT]
/ itlin`

In fact the parameters of linear research make it possible to ensure a better convergence of the method of NEWTON (cf [R5.03.01] for more details). The maximum iteration count is given `itelin` to carry out (the value by default 0 indicates that one does not make linear research) and precision `reslin` to reach to carry out the convergence of linear research.

Note:

It is not necessary to specify a precision nor an iteration count very high, the practice showing that 2 or 3 iterations of linear research are sufficient. One can thus be satisfied to ask for 3 iterations with the precision by default.

3.11 Keyword **MODELE_REDUIT**

Specify the characteristics of the method of resolution of the non-linear incremental problem by a method of reduction of model (see [R5.01.05]).

◇ `REAC_ITER = /0 [DEFECT]
/it`
◆ `BASE_PRIMAL = mode_empi`

```

◇  DOMAINE_REDUIT = / 'NOT' [DEFECT]
                        / 'YES'
    If DOMAINE_REDUIT = 'YES'
        ◆  GROUP_NO_INTERF = grno
    
```

Keywords MATRIX, PREDICTION, REAC_ITER and REAC_INCR the same significance and the same use as in the keyword factor have NEWTON (see §3.10).

It is necessary to provide an empirical base built on displacements (thanks to the operator DEFI_BASE_REDUITE). This base must be built on the same model and the same grid as calculation THER_NON_LINE.

The reduction of model is not compatible with the following features:

- With linear research
- With dualized limiting conditions (AFFE_CHAR_THER, it is necessary to use AFFE_CHAR_CINE)

It is possible to activate the very-reduction which uses a method of the type DEIM. In this case, calculation is tiny room on a zone of the grid restricted (called RID) and built using the operator DEFI_DOMAINE_REDUIT. It is necessary to give the group of nodes on which the interface using the keyword is defined GROUP_NO_INTERF.

3.12 Keyword CONVERGENCE

```
◇  CONVERGENCE :
```

Allows to define the values associated with the convergence criteria.

Note:

If none of the two operands following is present, then all occurs like if: RESI_GLOB_RELA = 1.E-6.

3.12.1 Operand RESI_GLOB_RELA

```
◇  RESI_GLOB_RELA = / 1.e-6
                        / testr
```

The algorithm continues the external iterations as long as the relative residue is higher than testr.

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

where F_i indicate the residue and S_i the thermal loading, the index n indicate $n^{ième}$ iteration.

3.12.2 Operand RESI_GLOB_MAXI

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

The algorithm continues the external iterations as long as the absolute residue is higher than testl.

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

where F_i indicate the residue, the index n indicate $n^{ième}$ iteration.

3.12.3 Operand ITER_GLOB_MAXI

◇ ITER_GLOB_MAXI = / 10
/ iterl

The algorithm continues the iterations as long as their number is lower than `iterl`.

3.13 Operand PARM_THETA

◇ PARM_THETA = `theta`

The argument `theta` is the parameter of the theta-method applied to the evolutionary problem. It must be ranging between 0 (explicit method) and 1 (completely implicit method). In the absence, keyword, the value used is $\theta=0.57$, a little higher than $\theta=0.5$ corresponds to the diagram of Crank-Nicholson. The incidence of the choice of `theta` on the stability of the method is detailed in [R5.02.02].

3.14 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].

3.15 Keyword FILING

◇ FILING =

This keyword is optional: by default, the whole of the computed fields for all the calculated steps is filed in the concept `result` resulting from the order. It is used to store certain sequence numbers in a structure of data `result` and/or to exclude from storage certain fields.

This keyword is identical to its equivalent for the operator `STAT_NON_LINE`, to refer to documentation [U4.51.03] for the description of under keywords.

Note:

In the event of stop of calculation by lack of time CPU, the steps of times previously calculated are saved in the base.

3.16 Keyword OBSERVATION

◇ OBSERVATION =

This keyword makes it possible post-to treat certain fields with the nodes or the elements on parts of model moments of a list (known as of observation) generally more refined than the list of the moments filed defined in the keyword `FILING` [§3.13] (where one stores all the fields on all the model). It is used primarily with economies as storage.

This keyword is répétable and allows the creation of a table of the same observation name than the concept `result` of `THER_NON_LINE`.

For the description of the syntax of this keyword factor, to refer to the documentation of the keyword are equivalent of `DYNA_NON_LINE` [U4.53.01].

3.17 Operand TITLE

◇ TITLE = `title`

Title which one wants to give to the result stored in `temper`, structure of data of the type `evol_ther` [U4.03.01].

4 Modeling

Problems of thermics nonlinear with models using the finite elements 3D, 2D or AXIS can be treated described in the documents [U3.22.01], [U3.23.01] and [U3.23.02] and [U3.24.01].

5 Example

One defined below the principal orders used to carry out a calculation of thermics not - linear transient. The example indicates how to continue calculation by enriching the concept result and how to specify the "initial" field.

```
lr8      =  DEFI_LIST_REEL  (... )

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

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

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

...

temple      =  THER_NON_LINE (  MODELE=moth, CHAM_MATER=chmat,
                               ETAT_INIT   =_F ( VALE     =  20.0),
                               INCREMENT   =_F ( LIST_INST =  lr8,
                                             NUME_FIN  =  20),
                               EXCIT      =_F ( LOAD    =  chth),
                               CONVERGENCE =_F ( RESI_GLOB_RELA =  1.E-8,
                                             ITER_GLOB_MAXI =  20),
                               )

...

temple =  THER_NON_LINE (  reuse =  temple,
                          MODELE=moth, CHAM_MATER=chmat,
                          ETAT_INIT   =_F ( EVOL_THER =  temple,
                                             NUME_ORDRE=  20),
                          INCREMENT   =_F ( LIST_INST =  lr8),
                          EXCIT      =_F ( LOAD    =  chth),
                          CONVERGENCE =_F ( RESI_GLOB_RELA =  1.E-6,
                                             ITER_GLOB_MAXI =  10 ),
                          )

...
```