
Operator THER_LINEAIRE

1 Drank

To solve a linear problem of thermal in steady or evolutionary mode.

The thermal loading is defined by key word `EXCIT`.

The temporal discretization of an evolutionary computation is provided by the list of times defined under key word `LIST_INST`. This computation can be initialized, at the first time, in three different ways (key word `ETAT_INIT`):

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

The product concept by this operator is of `evol_ther` type.

2 Syntax

```
temper [evol_ther] = THER_LINEAIRE
( reuse = temper,
  ◆MODELE=mo , [model]
  ◆CHAM_MATER=chmat , [cham_mater]
  ◇CARA_ELEM=carac , [cara_elem]
  ◆EXCIT=_F (
    ◆CHARGE=char , [load]
    FONC_MULT=fonc , [function, formula]
  ),
  ◆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",
  ),
  ◇INCREMENT=_F (
    ... to see STAT_NON_LINE [U4.51.03] key word INCREMENT ...
  ),
  ◇PARAM_THETA=/theta , [R]
    /0.57 , [DEFAULT]
  ◇SOLVEUR=_F ( ... to see [U4.50.01]... ),
  ◇ARCHIVAGE=_F (
    ... to see STAT_NON_LINE [U4.51.03] key word ARCHIVAGE ...
  ),
  ◇TITER=titer , [l_Kn]
)
```

3 Operands

3.1 MODEL Operand

◆MODELE = Mo

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

3.2 Operand CHAM_MATER

◆CHAM_MATER = chmat

Name of the affected material field on the model.

3.3 Operand CARA_ELEM

◇ CARA_ELEM = carac

the concept `carac` contains the characteristics of the shell elements thermal, if they exist in the model.

3.4 Key word EXCIT

◆EXCIT =

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

3.4.1 Operand CHARGE

◆CHARGE = tank

Concept of the type `charges` produces by `AFFE_CHAR_THER` or `AFFE_CHAR_THER_F` [U4.44.02]. As for the mechanics, it is also possible to define a load of type "imposed degree of freedom", here the temperature, by `AFFE_CHAR_CINE` [U4.44.03].

Notice important:

For each occurrence of the key word factor `EXCIT` the 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 the time (concept of type `function`, three-dimensions function 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_`.*

3.5 Key word ETAT_INIT

◇ETAT_INIT =

Makes it possible to define the initial field from which evolutionary computation is carried out.

Note:

If key word `ETAT_INIT` is absent, one carries out only steady computation at the time defined under key word `INCREMENT`.

The initial field is stored in data structure result `evol_ther` under sequence number 0.

3.5.1 Operand STATIONNAIRE

```
/STATIONNAIRE = "OUI"
```

the initial value of the field of temperature is then result of a preliminary steady computation.

3.5.2 Operand /VALE

```
VALE = tinit
```

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

3.5.3 Operand CHAM_NO

```
/CHAM_NO = tinit
```

the initial value is defined by a `cham_no` of temperature (result of operator `CREA_CHAMP` [U4.72.04]).

3.5.4 Operand EVOL_THER

```
/EVOL_THER = temp
```

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

3.5.5 Operand NUME_ORDRE/INST

```
◇/NUMÉRIQUE_ORDRE = nuini_evol  
/INST = instini_evol
```

Sequence number of the field to be extracted from this data structure. Extraction of the initial thermal state in `l'evol_ther_temp` starting from the number of archivage `NUME_ORDRE` or of the time of archivage `INST` to carry out the poursuite of computation. If `NUME_ORDRE` or `INSTS` are not filled, one takes the last existing number filed in `evol`.

Note:

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

3.5.6 Operand INST_ETAT_INIT

```
◇ INST_ETAT_INIT = istetaini
```

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

- when the initial state is defined by the data of the fields, associated time ago.
- when the state is given by a concept `evol_noli`, it is time in preceding computation (`istetaini = instini_evol`).

3.5.7 Operand PRECISION/CRITERE

cf [U4.71.00].

3.5.8 Operand STATIONNAIRE

```
/STATIONNAIRE = "OUI"
```

the initial value is that of a preliminary steady computation. That is possible only if the same mode of initialization is retained for the computation of the temperature.

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.

3.5.9 Operand EVOL_THER

/EVOL_THER = temp

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

3.5.10 Operand NUME_INIT

◇NUMÉRIQUE_INIT = nuinievol

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

3.6 Key word INCREMENT

◇INCREMENT =

Makes it possible to define times of computation which determine the intervals of time taken to integrate the differential equation.

The operands of key word INCREMENT have the same meaning as in operator STAT_NON_LINE, to see the document [U4.51.03].

Note:

If key word INCREMENT is absent, one creates a list of times reduced to the only reality 0 and one carries out a steady computation.

3.7 Operand PARM_THETA

◇PARM_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 of the key word, the value used is $\theta=0.57$, a little higher than $\theta=0.5$ corresponding with the diagram of Crank-Nicholson. The incidence of the choice of theta on the stability of the method is detailed in [R5.02.02].

3.8 Key word solver

◇SOLVEUR =

This key word factor is optional: it makes it possible to define the method of resolution of the linear systems.

This operand is common to all the global commands [U4.50.01].

3.9 Key word ARCHIVAGE

◇ARCHIVAGE =

This key word is optional: by default, all the computed fields for all the calculated steps is filed in the result concept resulting of the command. It is used for storing certain sequence numbers in a data structure result and/or excluding from storage certain fields.

This key word is identical to its equivalent for operator STAT_NON_LINE, to refer to documentation [U4.51.03] for the description of under key words.

Note:

In the event of stop of computation by lack of time CPU, time step previously calculated are saved in the base.

3.10 Operand TITER

◇TITER = title

Titrate that one wants to give to result the temp stored in data structure of the evol_ther type [U4.03.01].

4 Modelization

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

5 Example

5.1 Transient computation

```
lr8=DEFI_LISTE_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     ),
                        ),))

temple = THER_LINEAIRE  ( MODELE=moth      ,
                          CHAM_MATER=chmat ,
                          EXCIT=_F        ( CHARGE=chth      ),
                          ETAT_INIT=_F    ( STATIONNAIRE=' OUI' ),
                          INCREMENT=_F    ( LISTE_INST=lr8    ,
                                              NUME_INST_FIN=30   )
                        )

temple = THER_LINEAIRE  ( reuse=tempe     ,
                          MODELE=moth     ,
                          CHAM_MATER=chmat ,
                          EXCIT=_F        ( CHARGE=chth      ),
                          ETAT_INIT=_F    ( EVOL_THER=tempe  ,
                                              NUME_ORDRE=30     ),
                          INCREMENT=_F    ( LISTE_INST=lr8    ),
                        )
```

the first call to command THER_LINEAIRE makes it possible to carry out a steady computation at time 0. and to connect an evolutionary computation until time 0.1s (31 times of computation is 30 computations of evolution).

The second call makes it possible to enrich the concept temple precedent, evolutionary computation is continued as from the 31ième time of computation.

6 Notice

command CALC_CHAMP [U4.81.04] allows to calculate heat flux, at the points of integration or the nodes, the field at nodes of temperature thus obtained by THER_LINEAIRE.