Operator THER_LINEAIRE

1 Goal

To solve a linear problem of thermics in stationary or evolutionary mode.

The thermal loading is defined by the keyword EXCIT.

The temporal discretization of an evolutionary calculation is provided by the list of moments defined under the keyword LIST_INST. This calculation can be initialized, at the first moment, in three different ways (keyword ETAT_INIT):

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

The concept produced by this operator is of type evol_ther.
2 Syntax

temper [evol_ther] = THER_LINEAIRE

( reuse = temper,
  ♦ MODEL = Mo,
  ♦ CHAM_MATER = chmat,
  ♦ CARA_ELEM = carac,
  ♦ EXCIT = F(
      ♦ LOAD = tank,
      FONC_MULT = fonc,
  ),
  ♦ ETAT_INIT = F{
    / STATIONARY = ‘YES’,
    / VALE = tinit,
    / CHAM_NO = tinit,
    / EVOL_THER = temp,
    ♦ / NUME_ORDRE = nuini,
    / INST = instini,
    ♦ / PRECISION = /1.0E-3,
    / prec,
    ♦ / CRITERION = /‘RELATIVE’,
    / ‘ABSOLUTE’,
  ),
  ♦ INCREMENT = F{
    ... to see STAT_NON_LINE [U4.51.03] keyword INCREMENT ...
  },
  ♦ PARM_THETA = / theta,
    / 0.57,
  ♦ SOLVEUR = F ( ... to see [U4.50.01]...),
  ♦ FILING = F ( ... to see STAT_NON_LINE [U4.51.03] keyword FILING ...
  ),
  ♦ 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 Operand CARA_ELEM

◊ CARA_ELEM = carac

The concept carac contains the characteristics of the elements of thermal hull, if they exist in the model.

3.4 Keyword EXCIT

♦ EXCIT =

Operand 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 AFTE_CHAR_THER or by AFTE_CHAR_THER_F [U4.44.02]. As for mechanics, it is also possible to define a load of type “imposed degree of freedom”, here the temperature, by AFTE_CHAR_CINE [U4.44.03].

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_ *.

3.5 Keyword ETAT_INIT

◊ ETAT_INIT =

Allows to define the initial field from which evolutionary calculation is carried out.

Note:

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

The initial field is stored in the structure of data result evol_ther under the sequence number 0.
3.5.1 **Operand STATIONARY**

/ STATIONARY = 'YES'

The initial value of the field of temperature is then the result of a preliminary stationary calculation.

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 = tinit

The initial value is defined by one cham_no of temperature (result of the operator 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 in evol_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 (Istetaini = instini_evol).

3.5.7 **Operand PRECISION/CRITERE**

Cf [U4.71.00].

3.5.8 **Operand STATIONARY**

/ STATIONARY = 'YES'

The initial value is that of a preliminary stationary calculation. That is not possible that if the same mode of initialization is retained for the calculation of the temperature.
3.5.9 **Operand EVOL_THER**

/ EVOL_THER = temp

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

3.5.10 **Operand NUME_INIT**

◊ NUME_INIT = nuini_evol

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

3.6 **Keyword INCREMENT**

◊ INCREMENT =

Allows to define the moments of calculation which determine the time intervals taken to integrate the differential equation.

Operands of the keyword INCREMENT have the same meaning as in the operator STAT_NON_LINE, to see the document [U4.51.03].

**Note:** If the keyword INCREMENT is absent, one creates a list of moments reduced to the only reality 0 and one carries out a stationary calculation.

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 keyword, the value used is \( \theta = 0.57 \), a little higher than \( \theta = 0.5 \) corresponding 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].

3.8 **Keyword SOLVEUR**

◊ SOLVEUR =

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

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

3.9 **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.
3.10 **Operand TITLE**

◊ **TITLE = title**

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

4 **Modeling**

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

5.1 Transitory calculation

```plaintext
lr8 = DEFI_LIST_REEL ( BEGINNING = 0.E0,
                       INTERVAL = ( _F (JUSQU_A = 2.E-4 , NUMBER = 2 ),
                                  _F (JUSQU_A = 1.E-3 , NUMBER = 10 ),
                                  _F (JUSQU_A = 1.E-2 , NUMBER = 9 ),
                                  _F (JUSQU_A = 1.E-1 , NUMBER = 9 ),
                                  _F (JUSQU_A = 1.E0 , NUMBER = 9 ),
                                  _F (JUSQU_A = 2.0 , NUMBER = 10 ), )
                       )

temple = THER_LINEAIRE ( MODEL = moth,
                         CHAM_MATER = chmat,
                         EXCIT = _F ( LOAD = chth ),
                         ETAT_INIT = _F ( STATIONARY = 'YES'),
                         INCREMENT = _F ( LIST_INST = lr8,
                                         NUME_INST_FIN = 30 ) )

temple = THER_LINEAIRE ( reuse = temple,
                         MODEL = moth,
                         CHAM_MATER = chmat,
                         EXCIT = _F ( LOAD = chth ),
                         ETAT_INIT = _F ( EVOL_THER = temple
                                         NUME_ORDRE = 30 ),
                         INCREMENT = _F ( LIST_INST = lr8 ) )
```

The first call to the order THER_LINEAIRE allows to carry out a stationary calculation at moment 0.
and to connect an evolutionary calculation until the moment 0.1s (31 moments of calculation is 30 calculations of evolution).

The second call makes it possible to enrich the concept temple precedent, evolutionary calculation is
continued from the 31ème moment of calculation.

6 Notice

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