
Operator LIRE_CHAMP

1 Goal

To see a field in a file with format MED and to store it in a concept of type field.

The field is indicated in the file by its name and possibly by a temporal parameter.

The produced concept is of the type corresponding to what was required.

2 Syntax

```
champ_lu = LIRE_CHAMP (
    ◇ FORMAT = 'MED' , [DEFECT]
    ◆ TYPE_CHAM = / 'NOEU_TEMP_R' ,
                  / 'NOEU_DEPL_R' ,
                  / 'ELGA_SIEF_R' ,
                  / etc...
    ◆ GRID = my , [grid]
    ◆ NOM_MED = nommed , [TXM]
    ◇ # if TYPE-CHAM = 'ELxx_yyyy'
    ◇ MODEL = model [model]
    ◇ PROL_ZERO = / 'NOT' [DEFECT]
                  / 'YES'
    ◇ # Selection of the temporal parameter
    / NUME_ORDRE = order, [I]
    / NUME_PT = not of time, [I]
    / INST = inst, [R]
    ◇ / CRITERION = 'RELATIVE' [DEFECT]
    ◇ PRECISION = / PREC, [R]
                  / 1.0E-6, [DEFECT]
    / CRITERION = 'ABSOLUTE'
    ◆ PRECISION = PREC, [R]
    # Choice of the components to reading: the same ones as the field in
    ASTER
    or an imposed list
    ◆ / NOM_CMP_IDEM = 'YES' , [TXM]
    / ◆ NOM_CMP = lcmp , [1_TXM]
    ◆ NOM_CMP_MED = lcmpmed , [1_TXM]
    ◇ NOM_MAIL_MED = nomamed , [TXM]
    ◇ UNIT = / unit , [I]
            / 81 , [DEFECT]
    ◇ INFORMATION = / 1 , [DEFECT]
            [DEFECT]
            / 2 ,
)
```

3 Operands

3.1 Operand FORMAT

◇ FORMAT = 'MED'

Choice of the format of the file containing the field with reading.

Note:

Only the format MED is operational today. However, with the sequence of LIRE_TABLE then CREA_CHAMP/EXTR_TABLE one can read a field stored in a table with the format ASTER

3.2 Operand TYPE_CHAM

◆ TYPE_CHAM = / 'NOEU_TEMP_R',
/ 'NOEU_DEPL_R',
/ 'ELGA_SIEF_R',
/ etc...

One indicates the type of the concept here to be produced. The name of the type is built with usual logic Code_Aster. The first four characters are 'NOEU', 'ELEM', 'ELNO', 'ELGA' or 'CART'. One finds then '_'. The following sequence defines the type of field: 'TEMP', 'DEPL', 'SIEF', etc... The name ends in '_R', '_F' or '_C' according to the data-processing type of the values.

Example: 'NOEU_TEMP_R', 'NOEU_DEPL_R', 'ELGA_SIEF_R' etc...

Caution:

There is no consistency check! One can create a concept temperature very well by reading again a field which was a displacement with the writing of the file.

Note:

It can happen that values read in the file are not affected in the final field. For example, if one reads a field of pressure on elements TETRA4 whereas it must be affected on meshes of edge (because it is its nature), one will be informed by this kind of alarm:

```
<A> <LIRE_RESU> <LRCEME>  
NONAFFECTED VALUES IN THE FIELD: 3699  
VALUES READ IN THE FILE : 3699
```

3.3 Operand GRID

◆ GRID = my

Name of the grid ASTER on which the field with reading will be expressed.

3.4 Operand NOM_MED

◆ NOM_MED = nommed

Name according to convention MED of the field to reading in the file. It is a chain of 32 characters.

3.5 Operand MODEL

◇ MODEL = Mo

Name of the model ASTER on which the field with reading will be expressed. This operand is obligatory if the field with reading is a field "by elements" (TYPE_CHAM=' ELxx_YYYY')

3.6 Operand PROL_ZERO

◇ PROL_ZERO = 'NOT' / 'YES'

When one creates a field "by elements", the structure of this field is imposed by Aster. If for example, one creates a stress field "ELNO" on a model 3D, all the nodes of the elements must carry the components SIXX, SIYY, ..., SIYZ. If the field MED which one reads does not have all the values expected not Aster, these missing values should "be invented". The "invented" value will be 0. if PROL_ZERO=' OUI', it will be "Not" (Not has Number) if PROL_ZERO=' NON'

3.7 Selection of the temporal parameter NUME_ORDRE, NUME_PT, INST, CRITERION, PRECISION

If the field were written in the file without reference to a temporal parameter, nothing is to be mentioned in this reading command. If not it is necessary to specify about which moment it is. That is done by the designation of a sequence number, of step of time or a one moment value of filing. To refer to the document [U4.71.00] for the details on these keywords.

3.8 Operand NOM_CMP_IDEM or NOM_CMP and NOM_CMP_MED

The user must inevitably put NOM_CMP_IDEM or NOM_CMP in its command file.

3.8.1 Operand NOM_CMP_IDEM

◇ / NOM_CMP_IDEM = 'YES'

Indicate that one must read in file MED the components whose same name appears in the list components of the field within the meaning of *Code_Aster*.

3.8.2 Operands NOM_CMP, NOM_CMP_MED

◇ / ◆ NOM_CMP = lcmp,
◆ NOM_CMP_MED = lcmpmed,

These two lists must be of the same length. One reads in file MED the components listed in lcmpmed, then one affects them in the components within the meaning of *Code_Aster*, of the same row in the list lcmp.

3.9 Operand NOM_MAIL_MED

◇ NOM_MAIL_MED = nomamed

If this operand is absent, one seeks the field related to the first grid in the file. It is what occurs when the file contains one grid.

If the file contains several grids, one specifies here which is associated with the field that one wants to read.

3.10 Operand UNIT

◇ UNIT = unit

Number of the logical unit of the file, corresponds to the value provided in askt or by using the order DEFI_FICHIER.

Code_Aster

Version
default

Titre : Opérateur LIRE_CHAMP
Responsable : SELLENET Nicolas

Date : 02/11/2013 Page : 5/6
Clé : U7.02.02 Révision :
f15ef3cccf4

3.11 Operand INFORMATION

◇ INFORMATION = / 1, [DEFECT]
/ 2,

If INFORMATION is worth 2, some impressions of débogage take place.
If not, nothing takes place

4 Example

```
temp2 = LIRE_CHAMP (
    GRID      = m2,
    NOM_MED   = 'RESUUN__TEMP',
    NUME_ORDRE = 2,
    TYPE_CHAM = 'NOEU_TEMP_R',
)
```

This order will create a field with the nodes of name `temp2` and of type `NOEU_TEMP_R`. The grid support is `m2`. The values are those stored under the name `RESUUN__TEMP` in file `MED` provided on unit 81 with the sequence number 2.