Operator LIRE_RESU

1 Goal

To see fields and to store them in a structure of data of the type result

The type of the format of the file read is: either format MED, or universal format IDEAS, or format ENSIGHT.

Product a structure of data of the type result (evol_noli, evol_ther,...).

2 General syntax

```
resu = LIRE_RESU (

    # Choice of the format of the file to reading:

  ♦/FORMAT            =  'IDEAS',
     ♦  NOM_CHAM      =  l_nomch,                              [l_Kn]
     ◊  UNIT          =  /iunit,                               [I]
                         /19,                                  [DEFECT]
     ◊  FORMAT_IDEAS  = _F (…    # to see [ § 3.2.1.2] )

   /FORMAT            =  'IDEAS_DS58',
     ♦  UNIT          =  /inuit,                               [I]
                         /19,                                  [DEFECT]
     ♦  NOM_CHAM      =  l_nomch,                              [l_Kn]
     ◊  REDEFI_ORIENT=   _F (
                         ◊   CODE_DIR    =  /1,
                                            /2,
                                            /3,
                         ◊   DIRECTION   =  (dx, Dy, dz),     [l_R]
                         ◊   NODE        =  l_no,             [l_noeud]
                            ),
   /FORMAT            =  'ENSIGHT',
     ♦  NOM_CHAM      =  l_nomch,                              [l_Kn]
     ◊  NOM_FICHIER   =  file,                          [KN]

   /FORMAT            =  'MED'
     ◊  UNIT          =  /iunit,                               [I]
                         /81
     ♦  FORMAT_MED    =
                      (  _F (
                         ♦  NOM_CHAM      =  nomch,           [KN]
                         ♦  /NOM_CHAM_MED=   nommed,          [KN]
                            /NOM_RESU     =  nomres,          [KN]
                         ◊  NOM_CMP       =  lcmp,            [l_Kn]
                         ◊  NOM_CMP_MED   =  lcmpmed,         [l_Kn]
                      ), ),

    Is # Which structure of data necessary to create? :

    ♦ TYPE_RESU       =  /'EVOL_ELAS'
                         /'EVOL_CHAR'
                         /'EVOL_THER'
                         /'EVOL_NOLI'
                         /'DYNA_TRANS'
                         /'DYNA_HARMO'
                         /'MODE_MECA'
                         /'MODE_MECA_C'
                         /'MODE_EMPI'

  ♦  /MAILLAGE        =  my,                                   [grid]
     /MODELE          =  Mo,                                   [model]
  ◊  BEHAVIOR     =   _F (to see the document [U4.51.11]),
  ◊  CHAM_MATER       =  chmat,                                [cham_mater]
  ◊  CARA_ELEM        =  carac,                                [cara_elem]
  ◊  NB_VARI          =  nbvar,
```

 # C hoix of the sequence numbers to reading:

```
      ◊  /TOUT_ORDRE      =  'YES',                          [DEFECT]
         /NUME_ORDRE      =  lordr,                          [l_I]
         /LIST_ORDRE      =  lordr,                          [listis]
         /INST            =  linst,                          [l_R]
         /LIST_INST       =  linst,                          [listr8]
         /FREQ            =  lfreq,                          [l_R]
         /LIST_FREQ       =  lfreq,                          [listr8]
      ◊  |PRECISION       =  /prec,                          [R]
                             /1.D-06,                        [DEFECT]
         |CRITERION           =  /'RELATIVE',                   [DEFECT]
                             /'ABSOLUTE'

      ◊  TITLE             =  l_titre,                        [l_Kn]
      ◊  INFORMATION          =  /1,
                             /2
```

 # If TYPE_RESU=' EVOL_NOLI' :

```
      ◊  EXCIT =_F (
                    ♦  LOAD          =  tank,               [char_meca]
                    ◊  TYPE_CHARGE   =  /'FIXE_CSTE'
                                        /'FIXE_PILO'
                                        /'FIXE_PILO'
                                        /'SUIV'
                                        /'DIDI'
                    ◊  FONC_MULT     =  fonc,               [function]
                    ◊  DEPL          =  depl,               [function]
                    ◊  QUICKLY          =  quickly,             [function]
                    ◊  ACCE          =  acce,               [function]
                    ◊  MULT_APPUI    =  /'YES',
                                        /'NOT'              [DEFECT]
                    ◊  DIRECTION     =  (d1, d2, d3),       [l_R]
                    ◊  GROUP_NO      =  lgrno,              [l_gr_noeud]
                 ),
```

 # If TYPE_RESU= 'MODE_MECA' or 'MODE_MECA_C':

```
      ◊  MATR_RIGI =  matr_rigi,                             [matr_asse_DEPL_R]
      ◊  MATR_MASS =  matr_mass,                             [matr_asse_DEPL_R]
```

 # If TYPE_RESU= 'EVOL_THER' :

```
      ◊  EXCIT =_F (
                    ♦  LOAD          =  tank,               [char_ther]
                    ◊  FONC_MULT     =  fonc,               [function]
                  ),
```

 # If TYPE_RESU= 'EVOL_ELAS' :

```
       ◊  EXCIT =_F (
                    ♦  LOAD          =  tank,               [char_meca]
                    ◊  FONC_MULT     =  fonc,               [function]
                    ◊  TYPE_CHARGE   =  /typc               [l_Kn]
                                        /'FIXES'
                  ),
```

 # If TYPE_RESU= 'MODE_ EMPI '

```
      ◊  Digital_PLAN =  /0                                     [DEFECT]
                    /nume_plan,                            [I]
      )
```

3 Operands

3.1 Operands FORMAT / UNIT / NOM_FICHIER

/ FORMAT = 'IDEAS' or 'IDEAS_DS58'
Reading of the file to format IDEAS.

◊ UNIT
Logical number of unit of the file to the universal format IDEAS, by default 19.

/ FORMAT = 'MED'
Reading of the file to format MED.

◊ UNIT
Logical number of unit of the file to the format MED, by default 81.

/ FORMAT = 'ENSIGHT'
Reading of files to the Ensight format, relates to the reading of fields of pressure to the nodes and is associated with TYPE_RESU = 'EVOL_CHAR' and with NOM_CHAM = 'CLOSE'.

◊ NOM_FICHIER
File name (in small letters and between quotes) Ensight of the type 'Measured Results' (see Ensight To use Manual pages 3-49, 3-50) in particular containing:

- the name of the file 'Measured Geometry' containing of the numbers of nodes followed by their coordinates,
- the name of (or of) the file (S) 'Variable Scalar' containing the values of the pressure to the nodes specified in the file 'Measured Geometry'.

Note:

The logical number of unit can be associated with a file using the order DEFI_FICHIER [U4.12.03].

3.2 Operands if FORMAT = 'IDEAS'

They are not read *datasets* 58 (but only them *datasets* 55.57 and 2414)

3.2.1 Keyword FORMAT_IDEAS

3.2.1.1 Objective

The fields with reading in the universal file are written in the form of *datasets*. Each *dataset* is composed of a heading "*identity card*"and of a set of values (results with the nodes or by element with the nodes). This identity card is made up of several recordings "record", composed of fields"*field*". The objective of this keyword is to make it possible to the user to define his clean "identity card" by specifying its own search criteria.

Note:

A certain number of "identity cards" are defined by default [§6]. One can "overload them" by using the keyword FORMAT_IDEAS .
In certain typical cases (tests of not-regression) the grid associated with the model was not created by PRE_IDEAS. If one does not find in the grid the node associated with the number which appears in the dataset 55, then, one recovers the name of the node in comment in the universal file, while checking that the associated number corresponds well to the recovered name.

3.2.1.2 Syntax

```
FORMAT_IDEAS        =  _F (
♦  NOM_CHAM         =  nomch,            [KN]
◊  NUME_DATASET     =  /55,
                       /57,
                       /2414,
◊  |  RECORD_3      =  r3,               [l_I]
◊  |  RECORD_6      =  r6,               [l_I]
◊  |  RECORD_9      =  r9,               [l_I]
♦  POSI_ORDRE       =  Po,               [l_I]
◊  POSI_NUME_MODE   =  pnm,              [l_I]
◊  POSI_MASS_GENE   =  pmg,              [l_I]
◊  POSI_AMOR_GENE   =  pag,              [l_I]
◊  /  POSI_INST     =  pi,               [l_I]
   /  POSI_FREQ     =  PF,               [l_I]
♦  NOM_CMP          =  lcmp,             [l_Kn]
◊  NB_VARI          =  nbvari,           [I]
),
```

3.2.1.3 Operands

♦ NOM_CHAM = nomch

Reference symbol of the field for which the user defines the search criteria. See the keyword NOM_CHAM out of keywords factors [§3.5.1].

♦ NUME_DATASET =

Number of *dataset* from which will be extracted the results:

- : values with the nodes,
- : values with the nodes by element
- : values

◊ | RECORD_3
◊ | RECORD_6
◊ | RECORD_9

Each one of these keywords is composed of the word RECORD and of a number. The number indicates the number of the recording for which one will define the search criteria. Each operand makes it possible to define to the maximum 10 whole values.

Ex: RECORD_6 = (1, 4,9999,8,2,6),

In this example, if it *dataset* read contains on the level of the recording n°6 the values (1 4 9999 8 2 6), it will be retained for the continuation of research. Value 9999 is a joker making it possible to be unaware of the value read in *dataset*.

♦ / POSI_ORDRE

Vector of two entireties allowing to locate the sequence number

$V(1)$: Number of the recording

$V(2)$: Position of the sequence number

◊ / POSI_NUME_MODE

Vector of two entireties allowing to locate the number of mode

$V(1)$: Number of the recording

$V(2)$: Position of the number of mode

◊ / POSI_MASS_GENE

Vector of two entireties allowing to locate the generalized mass

$V(1)$: Number of the recording

$V(2)$: Position of the generalized mass

◊ / POSI_AMOR_GENE
Vector of two entireties allowing to locate generalized damping
$V(1)$: Number of the recording
$V(2)$: Position of generalized damping

◊ / POSI_INST
Vector of two entireties allowing to locate the moment
$V(1)$: Number of the recording
$V(2)$: Position of the moment

◊ / POSI_FREQ
Vector of two entireties allowing to locate the frequency
$V(1)$: Number of the recording
$V(2)$: Position of the frequency

♦ NOM_CMP
Name of the components to reading.

Ex: NOM_CMP = ('DX', 'DY', 'DZ', 'XXX', 'DRX', 'XXX', 'DRZ',),

The character string 'XXX' is a joker allowing to be unaware of the component during the reading of the values.
If the component count to reading is higher than the component count present in the file .unv, those are ignored.

Notice important:

When one is read cham_elem , this one is dimensioned in accordance with the finite elements of the model (see keyword MODEL below). For example, if one reads a stress field on a model 2D, the components carried by the elements will be SIXX , SIYY , SIXY and SIZZ . So in file IDEAS, one finds the components: SIXX , SIZZ , SIXZ , components SIXZ will be ignored. On the other hand, all components not found in the file (SIYY and SIXY in our example) will be put at zero.

Notice on the assignment of the field to the node in the datasets 55:

In a file with the universal format (Ideas), a node of the grid is known by its number. In Aster, a node is known by its name. The retranscription is done while assigning to the node, a name which starts with N follow-up of its Ideas number. During the reading of the field to the node in a dataset 55, one checks that the node exists in the grid Aster. If such is not the case, then one considers the name given, usually, in comment beside the number of the node in the dataset 55.

◊ NB_VARI
Many internal variables for the fields of the type ' VARI'

◊ PROL_ZERO =
If PROL_ZERO = 'YES' : one puts the values of the components at the nodes with 0 where the field is not defined, and one emits information of the type: "Nonexisting values of the field TEMP read on the grid given are considered worthless."
If PROL_ZERO = 'NOT' : one keeps the field such as it is.
This keyword is dedicated only to the fields with the nodes, because for the fields with the elements, the field is prolonged by 0 by default.

3.3 Operands if FORMAT = 'IDEAS_DS58'

Only the datasets 58 are read.

♦ NOM_CHAM = nomch,

3.3.1 Operand NOM_CHAM

Reference symbol of the fields to reading. One can read the fields with the nodes corresponding to: NOM_CHAM = 'DEPL', 'QUICKLY', 'ACCE', 'SIEF_NOEU', 'EPSI_NOEU'.

3.3.2 Keyword REDEFI_ORIENT

This keyword optional factor makes it possible to possibly redefine the significant direction of the sensor in certain points of measurement. This redefinition is treated currently only for NOM_CHAM = 'DEPL', 'QUICKLY' and 'ACCE'. One redefines as many orientations as necessary. If this keyword is absent, one adopts the usual convention recalled in the following table:

Code direction	'DEPL', 'QUICKLY', 'ACCE'	'SIEF_NOEU'	'EPSI_NOEU'
1	DX	SIXX	EPXX
2	DY	SIYY	EPYY
3	DZ	SIZZ	EPZZ
4		SIXY	EPXY
5		SIXZ	EPXZ
6		SIYZ	EPYZ

3.3.2.1 Operands

◊ CODE_DIR = /1,
/2,
/3,
Code describing the direction of measurement, given in the heading of the dataset 58.

◊ DIRECTION = (dx, Dy, dz)
Directing vector, expressed in the total reference mark, indicating the significant direction to associate with CODE_DIR

◊ NODE = l_no,
List of the nodes where CODE_DIR must be associated with the directing vector DIRECTION.

3.3.3 Note: structure of the field created

The universal files with the format dataset 58 contain fields "with hole": each dataset contains the data associated with a node and a direction. The field cannot be prolonged by zero values, as it is the case for the files of the type dataset 55.57 or 2414. One creates a field whose components are, for each node:

- Di , the number i being associated with the code direction in the local reference mark,
- DiX , DiY and DiZ , giving the local direction in the total reference mark.

Note: it is not possible to print and visualize directly the field read in the general formats (type MED). On the other hand, the field read can be used in operators of correlation calculation-test. The field result contains the standard components and can be printed.

3.4 Operands if FORMAT = 'MED'

With format MED it is not possible to read structures of data of the following type: DYNA_TRAN, DYNA_HARMO, MODE_MECA, MODE_MECA_C. However, it should be noted that this second reading does not make it possible to fill the structural parameters of data. Calculation can stop if these parameters are necessary to the execution of an order.

3.4.1 Operand NOM_CHAM

♦ NOM_CHAM = nomch,

Reference symbol of the field to reading. One can read the fields with the nodes a priori, the fields by elements with the nodes (ELNO), constant fields (ELEM) and of the fields at the points of integration (ELGA). See the keyword NOM_CHAM out of keywords factors [§3.5.1]

3.4.2 Operand NOM_CHAM_MED/NOM_RESU

♦ / NOM_CHAM_MED = nommed,
/ NOM_RESU = nommed,

Allows to define the name of field MED, either explicitly or in an automatic way:

- NOM_CHAM_MED : Name according to convention MED of the field to reading in the file. It is a chain of 32 characters,
- NOM_RESU : Prefix of the name of field MED to reading in the file. It is the name of the result associated with field MED. It is a chain of with more the 8 characters. It makes it possible to define the name of the automatic field med of way with the abundant data in NOM_CHAM.

3.4.3 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.
The names of components MED are limited to 16 characters.

If these 2 lists are absent, it is supposed that the names of the components are the same ones for MED and Aster.

3.4.4 Operand PROL_ZERO

During the reading of a field at the points of Gauss to format MED, one checks coherence between the points of Gauss such as they are defined in file MED and Aster.
If the number of points of Gauss “MED” is different amongst points from Gauss “Aster” one emits simply an alarm and one does not fill the field on the elements in question. One will thus have nondefinite values (Not) where the field is not defined. The keyword PROL_ZERO = 'YES' allows the user to put zero values if it wishes it.

3.5 Other operands

3.5.1 Operands TYPE_RESU/NOM_CHAM

♦ TYPE_RESU
Type of the structure of data result created.
For the type EVOL_VARC and for format MED, if the name of the field is 'IRRA'then NOMGD=' IRRA_R'.

♦ NOM_CHAM = l_nomch

Reference symbol of the fields to reading. It is under this (or these names) that the fields will be stored in the structure of data result.

A priori, for the format 'IDEAS', one can read the fields with the nodes (NOEU) or constant by element (ELEM) or by nodes with the elements (ELNO).

For the format 'ENSIGHT', one can read only the fields of pressure ('PRES') SD of the type evol_char.

For the format 'IDEAS_DS58', one does not treat for the moment the fields with the following nodes: displacement, speed, acceleration, constraint and deformation.

For the type 'EVOL_CHAR', the fields which one can read are:

NEAR	Fields with the nodes of pressure (N/m^2), component $PRES$
FVOL_3D	Fields with the voluminal nodes of forces (N/m^3), components FX, FY, FZ
FVOL_2D	Fields with the voluminal nodes of forces (N/m^3), components FX, FY
FSUR_3D	Fields with the surface nodes of forces (N/m^2), components FX, FY, FZ
FSUR_2D	Fields with the surface nodes of forces (N/m^2), components FX, FY
VITE_VENT	Fields with the nodes speed of the wind (m/s), components DX, DY, DZ
T_EXT	Map of outside temperature, component $TEMP$
COEF_H	Map of coefficient of exchange, component H
VECT_ASSE	Assembled vector of type DEPL_R

For the format 'MED', LE choice of the family of points of Gauss is done starting from the name provided by
the user, for example:

```
U3 =LIRE_RESU (  TYPE_RESU = ' EVOL_NOLI',
                 FORMAT    = ' MED', MODELE=MOMECA,
                 FORMAT_MED= (
                 _F (   NOM_CHAM_MED = ' U_______VARI_ELGA_____________ ',
                     NOM_CMP_MED  = ('V1', 'V2',),
                     NOM_CHAM     = ' VARI_ELGA'),)
              )
```

3.5.2 Operand NB_VARI

◊ NB_VARI = nbvar,

Many internal variables to reading for the fields of internal variables (VARI_R).

3.5.3 Operands MATR_RIGI/MATR_MASS

◊ MATR_RIGI = matr_rigi,
◊ MATR_MASS = matr_mass,

When one reads again a concept of the type mode_meca and that one wants to make use of it mode_meca in certain operators (for example PROJ_MATR_BASE), it is necessary that:

1) mode_meca product reference 2 matr_asse (rigidity and mass) which was used to calculate it,
2) fields of mode_meca are numbered same manner as the unknown factors from these 2 matr_asse.

So that this coherent classification is established in LIRE_RESU, these 2 optional keywords should be used MATR_RIGI and MATR_MASS (see for example the test sdnv102a).

Caution:

This possibility is available only to the format 'IDEAS'.

3.5.4 Operands GRID / MODEL

♦ / GRID = my,

Grid on which one affects the fields read.

/ MODEL = Mo,

Name of the model where are defined the types of finite elements affected on the grid. If one wants to read one cham_elem, it is necessary to give the name of the model.

3.5.5 Operand BEHAVIOR

The syntax of this keyword common to several orders is described in the document [U4.51.11]. This keyword must be indicated in the case of non-linear mechanics because it is used in recovery as calculation in STAT_NON_LINE and DYNA_NON_LINE to check the compatibility of the behaviors (many internal variables in particular). If it is not informed, the structure will be considered to have elastic behavior (COMPORTEMENT=' ELAS') in small deformations (RELATION=' PETIT').

The concept EVOL_THER is used with linear thermics and non-linear thermics. In this last case, the thermal behavior (keyword COMP_THER_NL in THER_NON_LINE) is not available in LIRE_RESU . What wants to say that the concept EVOL_THER created by LIRE_RESU does not contain the thermal map of behavior (COMPOR THER) and is thus not completely in conformity. However, this problem is not awkward insofar as the map of behavior is created in THER_NON_LINE, including in recovery, and that the non-linear thermal behaviors do not have internal variables, there is thus not necessary to check the compatibility of the non-linear fields in recovery, it is no risk of false results

3.5.6 Operands CHAM_MATER/CARA_ELEM/EXCIT

So that a result resulting from the order LIRE_RESU that is to say exploitable with Stanley, it is necessary to have the relative information for the material field and the elementary characteristics. By concern for completeness, information concerning the loading can also be stored in the structure of data result.

3.5.7 Operands TOUT_ORDRE/ NUME_ORDRE / LIST_ORDRE / INST / LIST_INST / FREQ / LIST_FREQ / PRECISION / CRITERION

Selection in a structure of data result [U4.71.00].

3.5.8 Reading of MODE_MECA

One can read clean modes stored with the format IDEAS. But to be able to re-use them in the operators of dynamics (in particular DYNA_TRAN_MODAL), one needs the assembled matrices (rigidity and mass) associated with these modes. Keywords MATR_RIGI and MATR_MASS (pointing out those of the operator of modal calculation CALC_MODES) allow to inform these two matrices.

3.5.9 Reading of MODE_EMPI

```
◊  Digital_PLAN =  /0                                    [DEFECT]
                   /nume_plan,                          [I]
```

The empirical modes on a thermal result can be defined on a section of the grid (model known as “linear”) for the simulation of welding. The operand NUME_PLAN allows to specify with which section appartiennent empirical modes read. If this parameter is worth zero, then the empirical mode read is 3D (case general).

3.5.10 In connection with an alarm

It can happen that the values read on the file cannot be recopied in the fields of SD_RESULTAT produced. For example, a field of pressure existing on surface meshs, can be recopied on a model containing only elements 3D. In this case, the code transmits a message of alarm resembling:

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

3.5.11 Operand TITLE

◊ TITLE

Title which one wants to give to the result [U4.03.01].

3.5.12 Operand INFORMATION

◊ INFORMATION = inf

Allows to print in the file MESSAGE related information with the unfolding of the operator.

4 Examples

4.1 Example 1: reading of one result of type 'dyna_trans'

One reads on the universal file IDEAS, fields of displacement, speed and acceleration at moments 1. , 2. , 3. , 4. and 5. ,

```
resu   =  LIRE_RESU (   FORMAT    =   'IDEAS',
                        MODEL     =   Mo,
                        TYPE_RESU =   'DYNA_TRANS',
                        NOM_CHAM  =   ('DEPL', 'QUICKLY', 'ACCE',),
                        INST      =   (1. , 2. , 3. , 4. , 5. ,),
                     )
```

4.2 Example 2: reading of one result of type 'evol_noli' by defining them search criteria

One reads on universal file IDEAS, the internal fields of variables and deformations at moment 15. by taking account of search criteria user.

```
INIT =LIRE_RESU (   MODEL         =  Mo,
                  FORMAT        =   'IDEAS',
                  TYPE_RESU     =   'EVOL_NOLI',
                  NOM_CHAM      =   ('VARI_ELNO', 'EPSA_ELNO'),
                  NB_VARI       =   2,
                  INST          =   15. ,
                  FORMAT_IDEAS  = ( _F (   NOM_CHAM     = 'VARI_ELNO',
                                       NUME_DATASET = 57,
                                       RECORD_6     = (1,4,3,9999,2,6),
                                       POSI_ORDRE   = (7.4,),
                                       POSI_INST    = (8 1)
                                       NOM_CMP      = ('V1', 'V2', 'V3',
'V4',)),
                                    _F (   NOM_CHAM     = 'EPSA_ELNO',
                                       RECORD_6     = (1,4,4,3,2,6),
                                       NOM_CMP      = ('EPXX', 'XXX', 'EPZZ',
                                                       'EPXY', 'EPXZ',
'EPYZ'))),
                )
```

Files IDEAS with reading

```
              -1
              57    %VALEURS AUX NOEUDS DES ELEMENTS
          ASTER  3.05.30 CONCEPT U CALCULE LE  - CHAMP PAR ELEMENT AUX NOEUDS DE NOM
          CHAMP PAR ELEMENT AUX NOEUDS DE NOM SYMBOLIQUE  VARI_ELNO_ELGA - VARI_1  (ELNO)
          ASTER  3.05.30 CONCEPT U CALCULE LE 29/12/95 A 09:56:55 DE TYPE EVOL_NOLI
          CHAMP PAR ELEMENT AUX NOEUDS DE NOM SYMBOLIQUE  VARI_ELNO_ELGA

Record 6 ->          1          4          3          0          2          6
Record 7 ->          2          1          1         (1)  <- POSI_ORDRE (7,4)
Record 8 ->  0.15000E+02  <- POSI_INST (8,1)
                     1          1          8          6      % MAILLE MA2
            2.07919E-05  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
            2.07919E-05  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
            2.07919E-05  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
            2.07919E-05  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
            2.07919E-05  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
            2.07919E-05  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
            2.07919E-05  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
            2.07919E-05  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
              -1
              -1
              57    %VALEURS AUX NOEUDS DES ELEMENTS
          ASTER  3.05.30 CONCEPT U CALCULE LE  - CHAMP PAR ELEMENT AUX NOEUDS DE NOM
          CHAMP PAR ELEMENT AUX NOEUDS DE NOM SYMB - EPXX EPXY EPYY EPXZ EPYZ EPZZ  (ELNO)
          ASTER  3.05.30 CONCEPT U CALCULE LE 29/12/95 A 09:56:55 DE TYPE EVOL_NOLI
          CHAMP PAR ELEMENT AUX NOEUDS DE NOM SYMBOLIQUE  EPSA_ELNO

                     1          4          4          3          2          6
                     2          1          1          1
            0.15000E+02
                     1          1          8          6      % MAILLE MA2
            0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
            0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
            0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
            0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
            0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
            0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
            0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
            0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
              -1
```

4.3 Example 3: reading of a result of the type `'evol_ther'` by defining the search criteria

One reads on universal file IDEAS, the field of temperature for moment 0.8 by taking account of search criteria user.

```
TEMP = LIRE_RESU (  GRID         =  e-mail,
                    UNIT         =   19,
                    FORMAT       =   'IDEAS',
                    TYPE_RESU    =   'EVOL_THER',
                    NOM_CHAM     =   'TEMP',
                    INST         =   0.8,
                    FORMAT_IDEAS = _F (  NOM_CHAM      =   'TEMP',
                                         NUME_DATASET  =   2414,
                                         RECORD_3      =   (1,),
                                         RECORD_9      =   (2,4,1,5,2,1),
                                         POSI_ORDRE    =   (10.7),
                                         POSI_INST     =   (12.1),
                                      ),
                 )
```

4.4 Example 4: reading of a result of the type 'evol_char ' with the format 'ENSIGHT'

One reads on file ENSIGHT all the fields of pressure

```
near   =  LIRE_RESU (  FORMAT       =  'ENSIGHT',
                       NOM_FICHIER  =  'face6.result',
                       MODEL        =  Mo,
                       TYPE_RESU    =  'EVOL_CHAR',
                       NOM_CHAM     =  'NEAR',
                       TOUT_ORDRE   =  'YES',
                    )
```

Files ENSIGHT with reading

```
FILE face6.result
******************************

         1        0        1
         3
    .50000E-03  .10000E-02 .15000E-02
         1       1
face6.geom
presface6. ***    pressure

FILE face6.geom
**************************

File Ensight measured geometry file
particles coordinates
         10
         13  1.48000e+03-7.77010e+027.42010e+02
         14  1.48000e+03-7.77010e+027.42010e+02
         15  1.48000e+03-7.77010e+027.42010e+02
         16  1.48000e+03-7.77010e+027.42010e+02
         17  1.48000e+03-7.77010e+027.42010e+02
         18  1.48000e+03-7.77010e+027.42010e+02
         19  1.48000e+03-7.77010e+027.42010e+02
         20  1.48000e+03-7.77010e+027.42010e+02
         21  1.48000e+03-7.77010e+027.42010e+02
         22  1.48000e+03-7.77010e+027.42010e+02

FILE presface6. ***
**********************

presface6.001
 .10000E+00  .10000E+00   .10000E+00   .10000E+00   .10000E+00
   .10000E+00
 .10000E+00  .10000E+00   .10000E+00

presface6.002
 .10000E+01  .10000E+01   .10000E+01   .10000E+01   .10000E+01
   .10000E+01
 .10000E+01  .10000E+01   .10000E+01

presface6.003
 .10000E+02  .10000E+02   .10000E+02   .10000E+02   .10000E+02
   .10000E+02
 .10000E+02  .10000E+02   .10000E+02
```

4.5 Example 5: reading of one evol_ther with format MED

```
LIRE_RESU (   FORMAT = 'MED', GRID = MY,
              UNIT = 21, TOUT_ORDRE = 'YES', TYPE_RESU = 'EVOL_THER',
              FORMAT_MED = _F (   NOM_CHAM = TEMP,
                                  NOM_CHAM_MED = 'THERDEP_TEMP',  )
           )
```

Annexe 1 : FORMAT_IDEAS : values by default

In this paragraph, we present for each field (NOM_CHAM) search criteria by default used to locate in the universal file the results to read.

CHAM_NO				
NOM_CHAM	'DEPL'	'QUICKLY'	'ACCE'	'TEMP'
NUME_DATASET	55	55	55	55
RECORD_3				
RECORD_6	1 4 3 8 2 6	1 4 3 11 2 6	1 4 3 12 2 6	2 4 1 5 2 1
RECORD_9				
POSI_ORDRE	7 4	7 4	7 4	7 4
POSI_INST	8 1	8 1	8 1	8 1
POSI_FREQ				
NOM_CMP	'DX' 'DY' 'DZ' 'DRX' 'DRY MARTINI' 'DRZ'	'DX' 'DY' 'DZ' 'DRX' 'DRY MARTINI' 'DRZ'	'DX' 'DY' 'DZ' 'DRX' 'DRY MARTINI' 'DRZ'	'TEMP' 'TEMP_MIL' 'TEMP_INF' TEMP_SUP'

CHAM_ELEM				
NOM_CHAM	'VARI_ELNO'	'EPSA_ELNO'	'SIEF_ELNO'	'CLOSE'
NUME_DATASET	57	57	57	57
RECORD_3				
RECORD_6	1 4 3 0 2 6	1 4 4 3 2 6	1 4 4 2 2 6	1 4 1 15 2 1
RECORD_9				
POSI_ORDRE	7 4	7 4	7 4	7 4
POSI_INST	8 1	8 1	8 1	8 1
POSI_FREQ				
NOM_CMP	'V1' 'V2' 'V3' 'V4' 'V5' 'V6' 'V7' 'V8' … … 'V9' 'V30'	'EPXX' 'EPXY' 'EPYY' 'EPXZ' 'EPYZ' 'EPZZ'	'SIXX' 'SIXY' 'SIYY' 'SIXZ' 'SIYZ' 'SIZZ'	'CLOSE'