Operator **LIRE_RESU**

1 Drank

Lira of the fields and to store in a data structure of the type `result`

the type of the format of the file read is: either med format, or the universal format IDEAS, or ensight format.

Product a data structure 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]
                                ◊UNITE=/iunit                    ,          [I]
                                                        /19,               [DEFAULT]

                                ◊  FORMAT_IDEAS=     _F  (…  #  to see [ § 3.2.1.2]  )

                             /FORMAT                     =   ' IDEAS_DS58',
                                ♦   UNITE=/inuit              ,          [I]
                                                        /19,               [DEFAULT]

                                ♦  NOM_CHAM         =l_nomch   ,          [l_Kn]
                                ◊REDEFI_ORIENT=_F              (
                                   ◊    CODE_DIR=/1        ,
                                                        /2,
                                                        /3,
                                   ◊    DIRECTION=       (dx, Dy, dz),       [l_R]
                                   ◊    NOEUD=l_no             ,
[l_noeud]
                                      ),

                             /FORMAT                     =   ' ENSIGHT',
                                ♦  NOM_CHAM=l_nomch           ,          [l_Kn]
                                 ◊NOM_FICHIER=fichier            ,          [kN]

                             /FORMAT                     =   `MED'
                                 ◊UNITE=/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]
                                ), ),
                          # Which data structure is necessary it to create?:

                          ♦   TYPE_RESU=/                     "EVOL_ELAS",
                                                        /   "EVOL_CHAR",
                                                        /   "EVOL_THER",
                                                        /   "EVOL_NOLI",
                                                        /   "DYNA_TRANS",
                                                         /  "MODE_MECA",
                                                        /   "MODE_MECA_C",

                          # If TYPE_RESU= "EVOL_NOLI":

                          ◊  NB_VARI=nbvar                 ,

                           # 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]
N

                                # different information:

                                ♦/MAILLAGE=ma                  ,                   [mesh]
                                   /MODELE                =mo    ,
[model]
                                ◊   COMP_INCR=_F               (see the document [U4.51.11]),
                                ◊   CHAM_MATER=chmat           ,                 [cham_mater]
                                ◊   AFFE_CARA_ELEM=carac          ,
[cara_elem]

                                 # choice of the sequence numbers to reading:

                                ◊   /TOUT_ORDRE          =   ' OUI',
[DEFAULT]
                                    /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,             [DEFAULT]
                                    |   CRITERE=/          "RELATIF",           [DEFAULT]
                                                           /"ABSOLU"

                                 # If TYPE_RESU= "EVOL_THER":

                                ◊   EXCIT=_F   (
                                    ♦   CHARGE=char               ,
[char_ther]
                                    ◊   FONC_MULT=fonc            ,
[function]
                                             ),

                                 # If TYPE_RESU= "EVOL_NOLI":

                                ◊   EXCIT=_F   (
                                    ♦   CHARGE=char               ,
[char_meca]
                                    ◊   TYPE_CHARGE=/      "FIXE_CSTE"
                                                           /"FIXE_PILO"
                                                           /"FIXE_PILO"
                                                           /"SUIV"
                                                           /"DIDI"
                                    ♦/     FONC_MULT=fonc         ,
[function]
                                       /   ♦DEPL=depl             ,
[function]
                                           ♦VITE=vite             ,
[function]
                                           ♦ACCE=acce             ,
[function]
                                       ◊MULT_APPUI=/       "OUI',
                                                           /"NON"            [DEFAULT]
```

```
                              ◊DIRECTION=          (d1, d2, d3),          [l_R]
                              ◊  /   NOEUD=lno            ,
[l_noeud]
                                 /   GROUP_NO=lgrno     ,
[l_gr_noeud]
                                     ),
```

# If `TYPE_RESU= "EVOL_ELAS"`:

```
                        ◊EXCIT=_F      (
                              ♦  CHARGE=char                ,
[char_meca]
                              ◊  FONC_MULT=fonc             ,
[function]
                              ◊TYPE_CHARGE     =    typc               [l_Kn]
                                                  /"FIXES"
                                     ),


                        ◊   TITER=l_titer              ,               [l_Kn]
                        ◊   INFO=/1                 ,
                                                  /2
                        )
```

3 Operands

3.1 Operands FORMAT / UNITE / NOM_FICHIER

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

◊UNITE
logical Number of unit of the file to the universal format IDEAS, by default 19.

/FORMAT = "MED"
Reading of the file to med format.

◊UNITE
logical Number of unit of the file to med format, by default 81.

/FORMAT = "ENSIGHT"
Reading of files to 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 = "NEAR".

◊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 command DEFI_FICHIER [U4.12.03].*

3.2 Operands if `FORMAT = "IDEAS"`

One fields to read does not read the datasets 58 (but only the datasets 55,57 and 2414

3.2.1) `Key word`

3.2.1.1 FORMAT_IDEAS

Purpose in the universal file are written in the form of *datasets*. Each *dataset* is composed of a heading "*identity card*" and a set of values (results with the nodes or by element with the nodes). This identity card is made up of several records "record", composed of fields "*field*". The purpose of this key word 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 them "overload" key word FORMAT_IDEAS* `by means of.`
> *In certain typical cases (tests of NON-regression) the mesh associated with the model was not created by* `PRE_IDEAS`*. If one does not find in the mesh 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]
◊NUMÉRIQUE_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_NUMÉRIQUE_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`
Symbolic name of the field for which the user defines the search criteria. See key word `NOM_CHAM` out of key keys factors [§3.5.1].

♦`NUMÉRIQUE_DATASET =`
Number of *the dataset* from which will be extracted the results:

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

◊ | `RECORD_3`
◊ | `RECORD_6`
◊ | `RECORD_9`
Each one of these keywords is composed of the key `RECORD` and a number. The number indicates the number of the record 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 *the dataset* read contains on the level of the record n°6 the values (1 4 9999 8 2 6), it will be retained for the continuation of the searches. Value 9999 is a joker making it possible to be unaware of the value read in *the dataset*.

♦ / `POSI_ORDRE`
Vector of two integers allowing to locate sequence number:
$V(1)$ Number of the record
$V(2)$: Position of sequence number

◊/`POSI_NUMÉRIQUE_MODE`
Vector of two integers allowing to locate the number of mode
$V(1)$: Number of the record
$V(2)$: Position of the number of mode

◊/`POSI_MASS_GENE`
Vector of two integers allowing to locate the generalized mass
$V(1)$: Number of the record
$V(2)$: Position of generalized mass

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

◊/`POSI_INST`
Vector of two integers allowing to locate time
$V(1)$: Number of the record
$V(2)$: Position of time

◊/`POSI_FREQ`
Vector of two integers allowing to locate the frequency
$V(1)$: Number of the record
$V(2)$: Position of 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 making it possible to be unaware of the component during the reading of the values.
So the number of components with reading is higher than number of components present in the file `.unv`, those are ignored.

Notice important:

> *When a cham_elem `is read` , this one is dimensioned in accordance with the finite elements of the model (see key word `MODELS` below). For example, if one 2D reads a stress field on a model, the components carried by the elements will be `SIXX` , `SIYY` , `SIXY` and `SIZZ` . So in the file IDEAS, one finds the components: `SIXX` , `SIZZ` , `SIXZ` , components `SIXZ` will be ignored. On the other hand, all the components not found in 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 mesh 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 Aster mesh. 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 local variables for the fields of the type " VARI"

◊PROL_ZERO =
If `PROL_ZERO = "OUI'` : one of the components puts the values at the nodes with 0 where the field is not defined, and one transmits an information of the type: "The nonexisting values of field `TEMP` read on the mesh given are considered null."
If `PROL_ZERO = "NON"` : one keeps the field such as it is.
This key word is dedicated only to the fields at nodes, because for the fields with the elements, the field is prolonged by 0 by default.

3.3 Operands if FORMAT = "IDEAS_DS58"

One reads only the datasets 58.

3.3.1 Operand NOM_CHAM

♦ NOM_CHAM =nomch ,

Symbolic name of fields to read. One can read the fields at nodes corresponding to:
`NOM_CHAM = "DEPL", "QUICKLY", "ACCE", "SIEF_NOEU", "EPSI_NOEU".`

3.3.2 Key word REDEFI_ORIENT

This key word 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 directional senses as necessary. If this key word 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,
```

Codes describing the direction of measurement, given in the heading of the dataset 58.

```
◊ DIRECTION = (dx, Dy, dz)
```

directing Vector, expressed in the total reference, indicating the significant direction to associate with CODE_DIR

```
◊ =l_no NOEUD,
```

List of the nodes where CODE_DIR must be associated with 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 the node is outside the field of definition with a right profile of the EXCLU type node:

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

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

3.4 Operands if **FORMAT = “MED”**

With med format it is not possible to read data structures of the following type: DYNA_TRAN, DYNA_HARMO, MODE_MECA, MODE_MECA_C. However, it should be noted that this relecture does not make it possible to fill the parameters of data structure. The computation can stop if these parameters are necessary to the execution of a command.

3.4.1 Operand **NOM_CHAM**

```
♦ NOM_CHAM =nomch   ,
```

Symbolic name of the field to read. One can read a priori the fields at nodes, the fields by elements with nodes (ELNO), the fields constant (ELEM) and of the fields at the points of integration (ELGA). See key word NOM_CHAM out of key keys factors [§3.5.1]

3.4.2 Operand **NOM_CHAM_MED/NOM_RESU**

```
♦ /   NOM_CHAM_MED = nommed,
  /   NOM_RESU = nommed,
```

Makes it possible 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 read in the file. It is a character string of 32 characters,
- NOM_RESU : Name of field MED to reading in the file prefixes. It is the name of result associated with field MED. It is a character string of with more the 8 characters. It makes it possible to define the name of the automatic field med of way with the data provided to 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 med file 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`.
Names of the components MED are restricted with 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 to Gauss points to med format, one checks coherence between Gauss points such as they are defined in med file and Aster.
If number of Gauss points “MED” is different from the number of Gauss points “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 key word `PROL_ZERO = "OUI'` makes it possible to the user to put zero values if it wishes it.

3.5 Other operands

3.5.1 Standard Operands

`TYPE_RESU/NOM_CHAM`
♦TYPE_RESU of data structure `result` created.
For type `EVOL_VARC` and med format, if the name of the field east “`IRRA`” then `NOMGD=' IRRA_R'`.

♦ `NOM_CHAM = l_nomch`

Symbolic name of fields to read. It is under this (or these names) that the fields will be stored in data structure `result`.

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

For format `"ENSIGHT"`, one can read only the fields of pressure (“`NEAR`”) of the SD of the evol_char `type`.

For format `"IDEAS_DS58"`, one does not treat for time the following fields at nodes: displacement, velocity, acceleration, stress and strain.

For format `"MED"`, the choice of the family of Gauss points 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 local variables to reading for the fields of local 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 mode_meca type and that one wants to make use of this mode_meca in certain operators (for example PROJ_MATR_BASE), it is necessary that:

1) the mode_meca produces reference the 2 matr_asse (stiffness and mass) which was used to calculate it,
2) fields of the mode_meca are numbered same way that the unknowns from these 2 matr_asse.

So that this coherent classification is established in LIRE_RESU, it is necessary to use these 2 optional key words 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 MAILLAGE / MODELS

```
♦/MAILLAGE=           my,
```

Mesh on which one affects the fields read.

```
   /MODELE       = Mo,
```

Name of the model where definite affected ones on the mesh the are element types finished. If one wants to read a cham_elem, it is necessary to give the name of the model.

3.5.5 Operand COMP_INCR

the syntax of this key word common to several commands is described in the document [U4.51.11]. This keyword must be indicated in the case of the nonlinear mechanics because it is used in recovery as computation in STAT_NON_LINE and DYNA_NON_LINE to check the compatibility of the behaviors (many local variables in particular). If it is not informed, the structure will be considered to have elastic behavior (COMP_ELAS=' ELAS') in small strains (RELATION=' PETIT').

Concept EVOL_THER is used with the thermal flax é area and the nonlinear thermal. 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 concept EVOL_THER created by LIRE_RESU does not contain the thermal card of behavior (COMPOR THER) and is thus not completely in conformity. However, this problem is not awkward insofar as the card of behavior is created in THER_NON_LINE , including in recovery, and that the nonlinear thermal behaviors do not have local variables, it is thus not necessary to check the compatibility of the nonlinear fields in recovery, there is no risk of results false

3.5.6 Operands CHAM_MATER/CARA_ELEM/EXCIT

So that result resulting of the command a LIRE_RESU is 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 data structure result.

3.5.7 Operands TOUT_ORDRE/ NUME_ORDRE / LIST_ORDRE / INST / LIST_INST / FREQ / LIST_FREQ / accuracy / CRITERE

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

3.5.8 Reading of the MODE_MECA

One can read eigen 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 matrixes (stiffness and mass) associated with these modes. Key words `MATR_RIGI` and `MATR_MASS` (pointing out those of the operators of modal computation `MODE_ITER_SIMULT` and `MODE_ITER_INV`) make it possible to inform these two matrixes.

3.5.9 In connection with an alarm

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

```
<A> <LIRE_RESU> <LRCEME>
    VALUES NON AFFECTED IN LE FIELD: 3699
    VALUES READ IN LE FICHIER        : 3699
```

3.5.10 Operand **TITER**

◊ `TITER`

Titrates that one wants to give to result [U4.03.01].

3.5.11 Operand **INFO**

◊ `INFO = inf`

Makes it possible to print in the message file of information related to the unfolding of the operator.

4 Examples

4.1 Example 1: reading of `result` of type "`dyna_trans`"

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

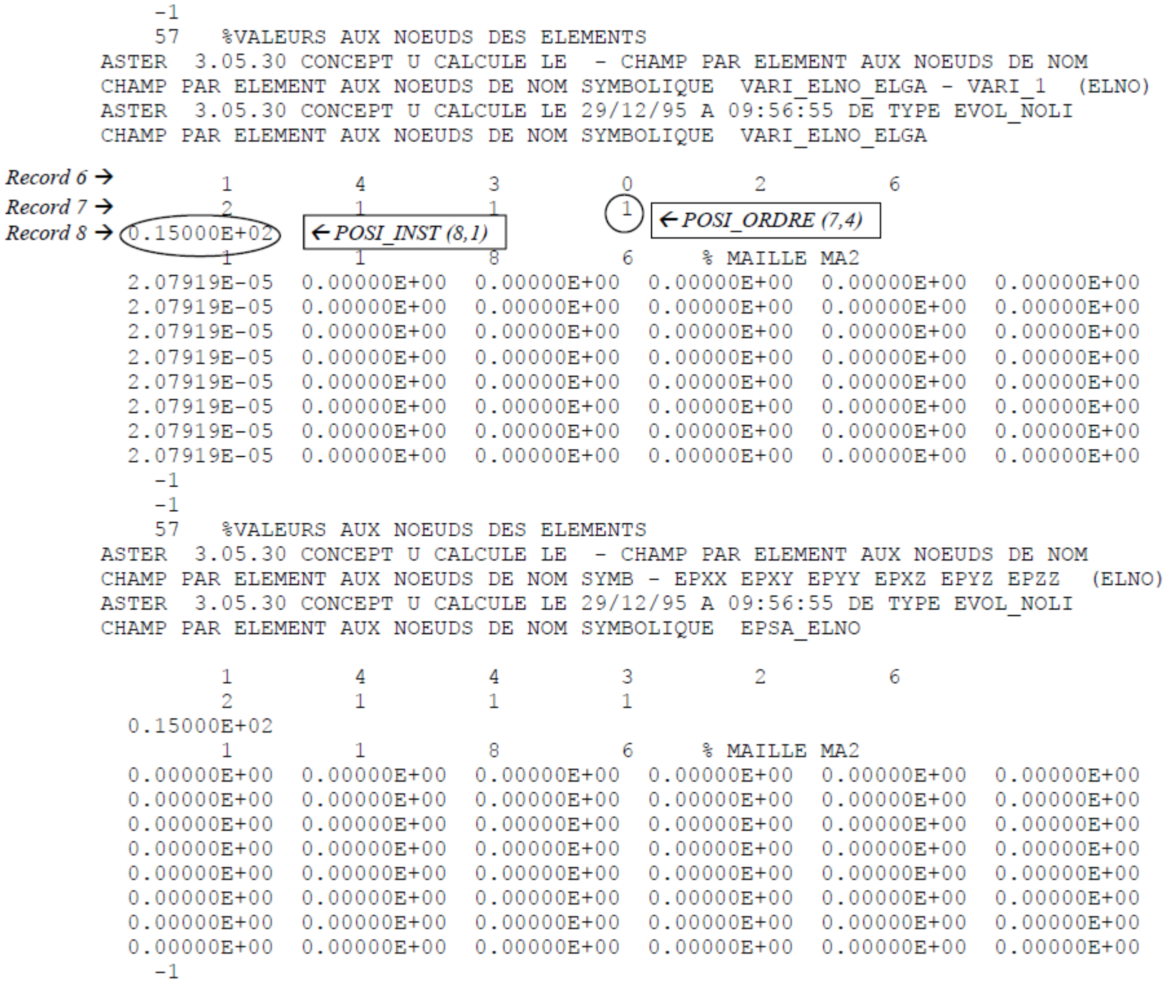
```
resu=LIRE_RESU          (           FORMAT=' IDEAS',
                       MODELE=mo        ,
                              TYPE_RESU=' DYNA_TRANS',
                       NOM_CHAM=     ("DEPL", "QUICKLY", "ACCE",),
                       INST=         (1. , 2. , 3. , 4. , 5. ,),
              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

4.2 Example 2: reading of `result` of type "`evol_noli`" by defining the search criteria

One reads on the universal file IDEAS, the fields of local variables and strains at time 15. by taking account of search criteria user.

```
INIT =LIRE_RESU (   MODELE=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 result of type “evol_ther” by defining the search criteria

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

```
TEMP =LIRE_RESU       ( MAILLAGE=mail       ,
                       UNITE=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 result of type "evol_char" to format "ENSIGHT"

One reads on file ENSIGHT all the fields of pressure

```
pres=LIRE_RESU          (              FORMAT=' ENSIGHT',
                              NOM_FICHIER=' face6.result',
                       MODELE=mo          ,
                                 TYPE_RESU=' EVOL_CHAR',
                                 NOM_CHAM=' PRES',
                              TOUT_ORDRE=' OUI',
                    )
```

Files ENSIGHT with reading

```
FICHIER face6.result
******************************

         1        0        1
         3
    .50000E-03  .10000E-02.15000E-02
         1       1
face6.geom
presface6. ***    pressure

FICHIER face6.geom
***************************

File Ensight measured geometry files
particles coordinates
         10
         131.48000e+03-7.77010e+027.42010e+02
         141.48000e+03-7.77010e+027.42010e+02
         151.48000e+03-7.77010e+027.42010e+02
         161.48000e+03-7.77010e+027.42010e+02
         171.48000e+03-7.77010e+027.42010e+02
         181.48000e+03-7.77010e+027.42010e+02
         191.48000e+03-7.77010e+027.42010e+02
         201.48000e+03-7.77010e+027.42010e+02
         211.48000e+03-7.77010e+027.42010e+02
         221.48000e+03-7.77010e+027.42010e+02

FICHIER 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 a `evol_ther` with med format

```
the LIRE_RESU  ( FORMAT = “MED”, MAILLAGE = MY,
                 UNITE = 21, TOUT_ORDRE = “OUI', TYPE_RESU = “EVOL_THER”,
                 FORMAT_MED = _F (   NOM_CHAM = TEMP,
                                     NOM_CHAM_MED = “THERDEP_TEMP”,  )
               )
```

Annexe 1 : FORMAT_IDEAS : default values

In this paragraph, we present for each field (NOM_CHAM) the search criteria per 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” “DRZ”	“DX” “DY” “DZ” “DRX” “DRY” “DRZ”	“DX” “DY” “DZ” “DRX” “DRY” “DRZ”	“TEMP” “TEMP_MIL” “TEMP_INF” TEMP_SUP'

CHAM_ELEM				
NOM_CHAM	“VARI_ELNO”	“EPSA_ELNO”	“SIEF_ELNO”	“NEAR”
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”	“NEAR”