

Manuel d'Utilisation
Fascicule U7.0- : Echanges de données
Document : U7.02.01

Opérateur *LIRE_RESU*

1 But

Lire des champs et les stocker dans une structure de données de type *resultat*. Les champs admis à la lecture doivent être de type champs aux nœuds ou par élément aux nœuds.

Le type du format du fichier lu est : soit le format universel IDEAS, soit le format ENSIGHT (seulement pour les champs de pression aux nœuds), soit le format MED pour les champs aux nœuds.

Produit une structure de données de type *resultat* (*evol_noli*, *evol_ther*, ...).

2 Syntaxe générale

```

resu = LIRE_RESU (

    # Choix du format du fichier à lire :

    ◇ /  FORMAT      =  'IDEAS' ,                                [DEFAULT]
        ◇ UNITE      =      /  iunit,                            [I]
        /  DATASET_58 =  'NON' ,                                [DEFAULT]
        ◇  FORMAT_IDEAS = _F ( ...      % voir [§ 3.2.2])

    /  DATASET_58      =  'OUI' ,
        ◇ REDEFI_ORIENT =  (_F(
            ◆ CODE_DIR   =  /1,
                                /2,
                                /3,
            ◆ DIRECTION  =  (dx,dy,dz),    [l_R]
            ◆ NOEUD      =  l_no,          [l_noeud]
            ),),
    /  FORMAT      =  'ENSIGHT' ,
    NOM_FICHIER    =  fichier,                [Kn]

    /  FORMAT      =  'MED' ,
    ◇ UNITE      =      /  iunit,                            [I]
        /  / 81,                                            [DEFAULT]
    ◆  NOM_MED = nommed ,                                [TXM]
    ◆  /  NOM_CMP_IDEM      =  'OUI' ,                    [TXM]
    /  ◆  NOM_CMP          =  lcmp,                        [l_TXM]
        ◆  NOM_CMP_MED    =  lcmpmed,                    [l_TXM]
    ◇  NOM_MAIL_MED = nomamed ,                        [TXM]

    # Quelle structure de données faut-il créer ? :

    ◆  /  TYPE_RESU =  'EVOL_ELAS' ,
        NOM_CHAM   =  'DEPL' ,

    /  TYPE_RESU =  'EVOL_THER' ,
        NOM_CHAM   =  'TEMP' ,

    /  TYPE_RESU =  'EVOL_CHAR' ,
        NOM_CHAM   =( |  'PRES' ,
                        |  'VITE_VENT' ,
                        |  'FVOL_3D' ,
                        |  'FSUR_3D' ,
                        |  'FVOL_2D' ,
                        |  'FSUR_2D' ),),

    /  TYPE_RESU =  'EVOL_NOLI' ,
        NOM_CHAM   =( |  'DEPL' ,
                        |  'VARI_ELNO' ,
                        |  'SIEF_ELNO' ,
                        |  'EPSA_ELNO' ),),
    ◇  NB_VARI=  nbvar,                                [I]

```

Titre : Opérateur *LIRE_RESU*
Auteur(s) : **J. PELLET, F. LEBOUVIER**

Date : 09/01/03
Clé : U7.02.01-D Page : 3/14

```

/  TYPE_RESU =      / 'DYNA_TRANS' ,
                        / 'DYNA_HARMO' ,
                        / 'HARM_GENE' ,
NOM_CHAM  = (      | 'DEPL' ,
                        | 'VITE' ,
                        | 'ACCE' ,
                        | 'SIEF_NOEU' ,
                        | 'EPSI_NOEU_DEPL' , ),
/  TYPE_RESU =      'MODE_MECA' ,
NOM_CHAM  =      'DEPL' ,
◇  MATR_A=      matr_rigi ,      [matr_asse_DEPL_R]
◇  MATR_B=      matr_mass ,      [matr_asse_DEPL_R]

# autres informations :
◇  /  MAILLAGE      =  ma ,      [maillage]
/  MODELE          =  mo ,      [modele]
◇  /  TOUT_ORDRE    =  'OUI' ,
/  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-03 , [DEFAULT]
                        |  CRITERE   =  /  'RELATIF' , [DEFAULT]
                        /  'ABSOLU' ,
◇  TITRE           =  l_titre ,    [l_Kn]
)
```

3 Opérandes

3.1 Opérandes **FORMAT / UNITE / NOM_FICHIER**

/ **FORMAT** = 'IDEAS'

Lecture du fichier au format IDEAS.

◇ **UNITE**

Numéro d'unité logique du fichier au format universel IDEAS, par défaut 19.

/ **FORMAT** = 'MED'

Lecture du fichier au format MED.

◇ **UNITE**

Numéro d'unité logique du fichier au format MED, par défaut 81.

/ **FORMAT** = 'ENSIGHT'

Lecture de fichiers au format Ensight, concerne la lecture de champs de pression aux nœuds et est associé à **TYPE_RESU** = 'EVOL_CHAR' et à **NOM_CHAM** = 'PRES'.

◇ **NOM_FICHIER**

Nom de fichier (en minuscules et entre quotes) Ensight de type 'Measured Results' (voir Ensight User Manuel pages 3-49, 3-50) contenant notamment :

- le nom du fichier 'Measured Geometry' contenant des numéros de nœuds suivis de leurs coordonnées,
- le nom du (ou des) fichier(s) 'Scalar Variable' contenant les valeurs de la pression aux nœuds spécifiés dans le fichier 'Measured Geometry'.

3.2 Opérandes si **FORMAT** = 'IDEAS'

3.2.1 Mot clé **DATASET_58** = 'OUI' / 'NON'

Ce mot clé permet de choisir le type des datasets IDEAS que l'on va lire. Soit on lit des datasets 55,57 et 2414 qui représentent des champs sur un maillage, soit on lit des datasets 58 qui représentent des évolutions temporelles sur un nœud d'un maillage.

/ 'NON' [DEFAULT]

On ne lit pas les datasets 58 (mais seulement les datasets 55,57 et 2414)

/ 'OUI'

On ne lit que les datasets 58 ; dans ce cas, on peut éventuellement redéfinir les directions de mesure en utilisant le mot clé **REDEFI_ORIENT**.

3.2.2 Mot clé **FORMAT_IDEAS**

3.2.2.1 Objectif

Les champs à lire dans le fichier universel sont écrits sous forme de datasets. Chaque dataset est composé d'un entête "carte d'identité" et d'un ensemble de valeurs (résultats aux nœuds ou par élément aux nœuds). Cette carte d'identité est composée de plusieurs enregistrements "record", composés de champs "field". L'objectif de ce mot clé est de permettre à l'utilisateur de définir sa propre "carte d'identité" en spécifiant ses propres critères de recherche.

Remarque :

| *Un certain nombre de "cartes d'identité" sont définies par défaut [§6]. On peut les "surcharger" en utilisant le mot clé **FORMAT_IDEAS**.*

3.2.2.2 Syntaxe

```
◇  FORMAT_IDEAS = ( _F(  ◆  NOM_CHAM      =  nomsym ,    [Kn]
                           ◆  NUME_DATASET =    /55 ,
                           /57 ,
                           /2414 ,
                           ◇  |  RECORD_3   =  r3 ,          [1_I]
                           |  RECORD_6   =  r6 ,          [1_I]
                           |  RECORD_9   =  r9 ,          [1_I]
                           ◆  POSI_ORDRE  =  po ,          [1_I]
                           ◆  /  POSI_INST =  pi ,          [1_I]
                           /  POSI_FREQ  =  pf ,          [1_I]
                           ◆  NOM_CMP    =  lcmp ,        [1_Kn]
                           ) , ) ,
```

3.2.2.3 Opérandes

- ◆ NOM_CHAM =
Nom symbolique du champ pour lequel l'utilisateur définit les critères de recherche.
- ◆ NUME_DATASET =
Numéro du dataset à partir duquel sera extrait les résultats :

- 55 : valeurs aux nœuds,
- 57 : valeurs aux nœuds par élément
- 2414 : valeurs

```
◇  |  RECORD_3
   |  RECORD_6
   |  RECORD_9
```

Chacun de ces mots clés est composé du mot `RECORD` et d'un nombre. Le nombre indique le numéro de l'enregistrement pour lequel on va définir les critères de recherche. Chaque opérande permet de définir au maximum 10 valeurs entières.

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

Dans cet exemple, si le dataset lu contient au niveau de l'enregistrement n°6 les valeurs (1 4 9999 8 2 6), il sera retenu pour la suite des recherches. La valeur 9999 est un joker permettant d'ignorer la valeur lue dans le dataset.

- ◆ / POSI_ORDRE
Vecteur de deux entiers permettant de localiser le numéro d'ordre

V(1) : N° de l'enregistrement
V(2) : Position du numéro d'ordre

- ◆ / POSI_INST
Vecteur de deux entiers permettant de localiser l'instant

V(1) : N° de l'enregistrement
V(2) : Position de l'instant

◆ / POSI_FREQ

Vecteur de deux entiers permettant de localiser la fréquence

V(1) : N° de l'enregistrement
V(2) : Position de la fréquence

◆ NOM_CMP

Nom des composantes à lire.

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

La chaîne de caractères 'XXX' est un joker permettant d'ignorer la composante lors de la lecture des valeurs.

Si le nombre de CMPS à lire est supérieur au nombre de CMPS présents dans le fichier .unv, celles-ci sont ignorées.

Remarque importante :

Lorsqu'on lit un `cham_elem`, celui-ci est dimensionné conformément aux éléments finis du modèle (voir mot clé `MODELE` ci-dessous). Par exemple, si on lit un champ de contraintes sur un modèle 2D, les composantes portées par les éléments seront `SIXX`, `SIYY`, `SIXY` et `SIZZ`. Si dans le fichier `IDEAS`, on trouve les CMPS : `SIXX`, `SIZZ`, `SIXZ`, les CMPS `SIXZ` seront ignorées. En revanche, toutes les CMPS non trouvées dans le fichier (`SIYY` et `SIXY` dans notre exemple) seront mises à zéro.

3.2.3 Mot clé REDEFI_ORIENT

Ce mot clé facteur permet de redéfinir éventuellement la direction sensible du capteur en certains points de mesure. Cette redéfinition n'est traitée actuellement que pour les `NOM_CHAM = 'DEPL'`, `'VITE'` et `'ACCE'`. On redéfinit autant d'orientations que nécessaire. Si ce mot clé est absent, on adopte la convention habituelle rappelée dans le tableau suivant :

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

3.2.3.1 Opérandes

- ◆ `CODE_DIR = /1,`
 `/2,`
 `/3,`

Code décrivant la direction de mesure, donné dans l'en tête du dataset 58

- ◆ `DIRECTION = (dx, dy, dz),`

Vecteur directeur, exprimé dans le repère global, indiquant la direction sensible à associer à `CODE_DIR`

- ◆ `NOEUD = l_no,`

Liste des nœuds où `CODE_DIR` doit être associée au vecteur directeur `DIRECTION`.

3.3 Opérandes si **FORMAT = 'MED'**

3.3.1 Opérande **NOM_MED**

- ♦ **NOM_MED** = *nommed*,
Nom selon la convention MED du champ à lire dans le fichier. C'est une chaîne de 32 caractères.

3.3.2 Opérande **NOM_CMP_IDEM**

- ♦ / **NOM_CMP_IDEM** = 'OUI' ,
Indique qu'on doit lire dans le fichier MED les composantes dont le même nom apparaît dans la liste des composantes du champ au sens du *Code_Aster*.

3.3.3 Opérandes **NOM_CMP, NOM_CMP_MED**

- ♦ /
 - ♦ **NOM_CMP** = *lcmp*,
 - ♦ **NOM_CMP_MED** = *lcmpmed*,Ces deux listes doivent être de même longueur. On lit dans le fichier MED les composantes listées dans *lcmpmed*, puis on les affecte dans les composantes au sens de *Code_Aster*, de même rang dans la liste *lcmp*.

3.3.4 Opérande **NOM_MAIL_MED**

- ◇ **NOM_MAIL_MED** = *nomamed*,
Si cet opérande est absent, on cherche le champ lié au premier maillage dans le fichier. C'est ce qui se passe lorsque le fichier ne contient qu'un seul maillage.
Si le fichier contient plusieurs maillages, on précise ici lequel est associé au champ que l'on veut lire.

3.3.5 Opérandes **TYPE_RESU / NOM_CHAM**

- ♦ **TYPE_RESU**
Type de la structure de données *resultat* créée.
- ♦ **NOM_CHAM**
Nom symbolique du ou des champs lus :

nom du champ	type	signification
'ACCE'	<i>cham_no</i>	accélérations
'DEPL'	<i>cham_no</i>	déplacements
'EPSA_ELNO'	<i>cham_elem</i>	déformations anélastiques aux nœuds par éléments
'EPSI_NOEU_DEPL'	<i>cham_no</i>	déformations aux nœuds
'FSUR_2D'	<i>cham_no</i>	forces "surfiques" (N/m2) en 2D (FX, FY)
'FSUR_3D'	<i>cham_no</i>	forces "surfiques" (N/m2) en 3D (FX, FY, FZ)
'FVOL_2D'	<i>cham_no</i>	forces "volumiques" (N/m3) en 2D (FX, FY)
'FVOL_3D'	<i>cham_no</i>	forces "volumiques" (N/m3) en 3D (FX, FY, FZ)
'PRES'	<i>cham_no</i>	pressions
'SIEF_ELNO'	<i>cham_elem</i>	contraintes aux nœuds par éléments
'SIEF_NOEU'	<i>cham_no</i>	contraintes aux nœuds
'TEMP'	<i>cham_no</i>	températures
'VARI_ELNO'	<i>cham_elem</i>	variables internes aux nœuds par éléments
'VITE'	<i>cham_no</i>	vitesse
'VITE_VENT'	<i>cham_no</i>	vitesse du vent : sert de chargement linéique dans les opérateurs <i>STAT_NON_LINE</i> et <i>DYNA_NON_LINE</i>

◇ `NB_VARI = nbvar,`

Nombre de variables internes à lire pour les champs de variables internes (`VARI_R`).

3.3.6 Opérandes `MAILLAGE` / `MODELE`

◆ `/ MAILLAGE = ma,`

Maillage sur lequel on affecte le ou les champs lus.

`/ MODELE = mo,`

Nom du modèle où sont définis les types d'éléments finis affectés sur le maillage. Si on veut lire un `cham_elem`, il faut donner le nom du modèle.

3.3.7 Opérandes `TOUT_ORDRE` / `NUME_ORDRE` / `LIST_ORDRE` / `INST` / `LIST_INST` / `FREQ` / `LIST_FREQ` / `PRECISION` / `CRITERE`

Sélection dans une structure de données `resultat` [U4.71.00].

3.3.8 Lecture des `MODE_MECA`

On peut lire des modes propres stockés au format `IDEAS`. Mais pour pouvoir les réutiliser dans les opérateurs de dynamique (en particulier `DYNA_TRAN_MODAL`), on a besoin des matrices assemblées (rigidité et masse) associées à ces modes. Les mots clés `MATR_A` et `MATR_B` (rappelant ceux de `MODE_ITER_SIMULT`) permettent de renseigner ces deux matrices.

Derrière `MATR_A`, on donne le nom de la matrice de rigidité, derrière `MATR_B` celui de la matrice de masse.

3.3.9 Opérande `TITRE`

◇ `TITRE`

Titre que l'on veut donner au résultat [U4.03.01].

4 Tableau des possibilités

Dans ce paragraphe on présente pour chacun des formats, le type de résultats et le nom des champs disponibles.

Type de résultats	DEPL	VITE	ACCE	PRES	TEMP	VARI_ELNO	EPSA_ELNO	SIEF_ELNO
EVOL_ELAS	I/M							
EVOL_THER					I/M			
EVOL_CHAR				I / E				
EVOL_NOLI	I/M					I	I	I
DYNA_TRANS	I/M	I/M	I/M					
DYNA_HARMO	I/M	I/M	I/M					
HARM_GENE	I/M	I/M	I/M					

* I → IDEAS ; E → ENSIGHT ; M → MED

Tableau 4-1 : Types de résultat et types de champ disponibles en fonction du format

Type de résultats	FVOL_3D	FVOL_2D	FSUR_3D	FSUR_2D	VITE_VENT	EPSI_NOEU_DEPL	SIEF_NOEU
EVOL_CHAR	I/M	I/M	I/M	I/M	I/M		
DYNA_TRANS						I58	I58
DYNA_HARMO						I58	I58
HARM_GENE						I58	I58

* I → IDEAS ; M → MED ; I58 → IDEAS + DATASET_58='OUI'

Tableau 4-2 : Types de résultat et types de champ disponibles en fonction du format (suite)

5 Exemples

5.1 Exemple 1 : lecture d'un resultat de type 'dyna_trans'

On lit sur le fichier universel IDEAS, les champs de déplacement, de vitesse et d'accélération aux instants 1., 2., 3., 4. et 5.,

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

5.2 Exemple 2 : lecture d'un resultat de type 'evol_noli' en définissant les critères de recherche

On lit sur le fichier universel IDEAS, les champs de variables internes et de déformations à l'instant 15. en tenant compte de critères de recherche utilisateur.

```
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' ) ) ) ,  
                                          )
```

Titre : Opérateur LIRE_RESU
Auteur(s) : J. PELLET, F. LEBOUVIER

Date : 09/01/03
Clé : U7.02.01-D Page : 11/14

Fichiers IDEAS à lire

```

-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      ①
Record 8 → 0.15000E+02  ← POSI_INST (8,1)  ← POSI_ORDRE (7,4)

      1      4      3      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
-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

```

5.3 Exemple 3 : lecture d'un résultat de type 'evol_ther' en définissant les critères de recherche

On lit sur le fichier universel IDEAS, le champ de température pour l'instant 0.8 en tenant compte de critères de recherche utilisateur.

```

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) ,
                                       ) ,
                    )

```

```

-1
2414
36
B.C. 0, TIME = 0.8, TEMPERATURE_36
Record 3 → 1
ASTER 5.01.00 CONCEPT TEMPE CALCULE - CHAMP AUX NOEUDS DE ...
CHAMP AUX NOEUDS DE NOM SYMBOLIQUE TEMP - TEMP
ASTER 5.01.00 CONCEPT TEMPE CALCULE LE 18/12/98 A 15:19:49 DE ...
CHAMP AUX NOEUDS DE NOM SYMBOLIQUE TEMP
NUMERO D'ORDRE: 35 INST: 8.00000E-01
Record 9 → 2 4 1 5 2 1
Record 10 → -11 0 1 0 1 0
Record 11 → 1500 0
Record 12 → 8.00000E-01 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.
Record 13 → 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.
1
2.00000E+02
-----
205
1.00000E+02
-1

```

Annotations:

- Record 3 →** points to the first record.
- Record 9 →** points to the header line with 6 columns of integers.
- Record 10 →** points to the first data line.
- Record 11 →** points to the second data line.
- Record 12 →** points to the third data line, where the value **8.00000E-01** is circled.
- Record 13 →** points to the fourth data line.
- POSI_ORDRE (10,7)** points to the value **35** in the header line.
- POSI_INST (12,1)** points to the value **8.00000E-01** in the first data line.

5.4 Exemple 4 : lecture d'un résultat de type 'evol_char' au format 'ENSIGHT'

On lit sur le fichier ENSIGHT tous les champs de pression

```

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

```

Fichiers ENSIGHT à lire

FICHIER face6.result

```

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

```

FICHIER face6.geom

Fichier Enight 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

```

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

```

5.5 Exemple 5 : lecture d'un evol_ther au format MED

```

THERDEP=LIRE_RESU(  FORMAT='MED', MAILLAGE=MA,
                    NOM_CHAM='TEMP', TYPE_RESU='EVOL_THER',

                    NOM_MED='THERDEP_TEMP_____',
                    NOM_CMP='TEMP',
                    NOM_CMP_MED='TEMP',
                    TOUT_ORDRE='OUI',
                    UNITE=21)

```

6 ANNEXE : FORMAT_IDEAS : valeurs par défaut

Dans ce paragraphe, nous présentons pour chaque champ (NOM_CHAM) les critères de recherche par défaut utilisés pour localiser dans le fichier universel les résultats à lire.

CHAM_NO				
NOM_CHAM	'DEPL'	'VITE'	'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_INF' TEMP_SUP'

CHAM_ELEM				
NOM_CHAM	'VARI_ELNO'	'EPSA_ELNO'	'SIEF_ELNO'	'PRES'
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'	'PRES'