

**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`

Le type du format du fichier lu est : soit le format universel IDEAS, soit le format ENSIGHT, soit le format MED.

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' ,
      NOM_CHAM      = l_nomch ,                      [l_Kn]
      ◇ UNITE        =      / iunit,                  [I]
                        / 19,                          [DEFAULT]
      ◇ FORMAT_IDEAS = _F ( ...      # voir [§ 3.2.2])

    / FORMAT      = 'IDEAS_DS58' ,
      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,                          [DEFAULT]
      ♦ FORMAT_MED =
        ( _F(
          ♦ NOM_CHAM    = nomch ,                      [Kn]
          ♦ NOM_CHAM_MED = nommed ,                    [Kn]
          ♦ / NOM_CMP_IDEM = 'OUI' ,
            / ♦ NOM_CMP    = lcmp,                      [l_Kn]
              ♦ NOM_CMP_MED = lcmpmed,                  [l_Kn]
          ◇ NOM_MAIL_MED = nomamed ,                    [Kn]
        ),),

    # Quelle structure de données faut-il créer ? :

    ♦ TYPE_RESU =
      / 'EVOL_ELAS' ,
      / 'EVOL_CHAR' ,
      / 'EVOL_THER' ,
      / 'EVOL_NOLI' ,
      / 'DYNA_TRANS' ,
      / 'DYNA_HARMO' ,
      / 'HARM_GENE' ,
      / 'MODE_MECA' ,

    # Si TYPE_RESU= 'EVOL_NOLI' :
      ◇ NB_VARI= nbvar,                                [I]

    # Si TYPE_RESU= 'MODE_MECA' :
      ◇ MATR_A= matr_rigi ,                            [matr_asse_DEPL_R]
      ◇ MATR_B= matr_mass ,                            [matr_asse_DEPL_R]

```

## # autres informations :

```
♦ / MAILLAGE = ma , [maillage]
  / MODELE = mo , [modele]
```

## # choix des numéros d'ordre à lire :

```
◇ / 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-03 , [DEFAULT]
  | CRITERE = / 'RELATIF' , [DEFAULT]
  | 'ABSOLU' ,
◇ TITRE = l_titre , [l_Kn]
```

)

## 3 Opérandes

### 3.1 Opérandes **FORMAT / UNITE / NOM\_FICHIER**

/ **FORMAT** = 'IDEAS' ou 'IDEAS\_DS58'

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 Enight, 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) Enight de type 'Measured Results' (voir Enight 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'.

**Remarque :**

| *Le numéro d'unité logique peut avoir été associé à un fichier à l'aide de la commande **DEFI\_FICHIER** [U4.12.03].*

### 3.2 Opérandes si **FORMAT** = 'IDEAS'

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

#### 3.2.1 Mot clé **FORMAT\_IDEAS**

##### 3.2.1.1 Objectif

Les champs à lire dans le fichier universel sont écrits sous forme de datasets. Chaque dataset est composé d'un en-tête "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**.*

**3.2.1.2 Syntaxe**

```

◇   FORMAT_IDEAS = ( _F(
                        ◆   NOM_CHAM      =   nomch ,      [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]
                        ) , ) ,

```

**3.2.1.3 Opérandes**

- ◆ NOM\_CHAM = nomch

Nom symbolique du champ pour lequel l'utilisateur définit les critères de recherche. Voir le mot clé NOM\_CHAM hors des mots clés facteurs [§3.5.1].

- ◆ 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 `SIXX` et `SIXZ` seront ignorées. En revanche, toutes les CMPS non trouvées dans le fichier (`SIYY` et `SIXY` dans notre exemple) seront mises à zéro.*

### 3.3 Opérandes si `FORMAT = 'IDEAS_DS58'`

On ne lit que les datasets 58.

On ne lit que les champs correspondant à :

NOM\_CHAM= 'DEPL', 'VITE', 'ACCE', 'SIEF\_NOEU', 'EPSI\_NOEU\_DEPL'.

#### 3.3.1 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

##### 3.3.1.1 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.

## 3.4 Opérandes si **FORMAT** = 'MED'

### 3.4.1 Opérande **NOM\_CHAM**

♦ `NOM_CHAM = nomch,`

Nom symbolique du champ à lire. On peut lire a priori les champs aux nœuds, les champs par éléments aux nœuds (**ELNO**) ou constants (**ELEM**). Voir le mot clé **NOM\_CHAM** hors des mots clés facteurs [§3.5.1]

### 3.4.2 Opérande **NOM\_CHAM\_MED**

♦ `NOM_CHAM_MED = nommed,`

Nom selon la convention MED du champ à lire dans le fichier. C'est une chaîne de 32 caractères.

### 3.4.3 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*.

### 3.4.4 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`.

Les noms des composantes MED sont limités à 16 caractères.

### 3.4.5 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.

## 3.5 Autres opérandes

### 3.5.1 Opérandes **TYPE\_RESU / NOM\_CHAM**

♦ `TYPE_RESU`

Type de la structure de données `resultat` créée.

♦ `NOM_CHAM = l_nomch`

Ce mot clé est obligatoire pour tous les formats sauf pour le format MED où il est interdit.

Nom symbolique du ou des champs à lire. C'est sous ce (ou ces noms) que les champs seront stockés dans la structure de données `resultat`.

A priori, pour le format 'IDEAS', on peut lire les champs aux nœuds (**NOEU**) ou constant par élément (**ELEM**) ou par nœuds aux éléments (**ELNO**). Pour un type de structure de données `resultat`, la liste des champs possibles est donnée dans les documents du fascicule [U5].

Pour le format 'ENSIGHT', on ne sait lire que les champs de pression ('PRES') des SD de type evol\_char.

Pour le format 'IDEAS\_DS58', on ne traite pour l'instant les champs aux nœuds suivants : déplacement, vitesse, accélération, contrainte et déformation.

### 3.5.2 Opérande NB\_VARI

◇ NB\_VARI = nbvar,

Nombre de variables internes à lire pour les champs de variables internes (VARI\_R).

### 3.5.3 Opérandes MATR\_A et MATR\_B

◇ MATR\_A= matr\_rigi ,

◇ MATR\_B= matr\_mass ,

Lorsque l'on relit un concept de type mode\_meca et que l'on veut se servir de ce mode\_meca dans certains opérateurs (par exemple PROJ\_MATR\_BASE), il faut que :

- le mode\_meca produit référence les 2 matr\_asse (rigidité et masse) qui ont servi à le calculer,
- les champs du mode\_meca soient numérotés de la même manière que les inconnues de ces 2 matr\_asse.

Pour que cette numérotation cohérente soit établie dans LIRE\_RESU, il faut utiliser ces 2 mots clé facultatifs MATR\_A et MATR\_B (voir par exemple le test sdnv102a).

#### Attention :

| Cette possibilité n'est disponible qu'au format 'IDEAS'.

### 3.5.4 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.

### 3.5.5 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].

### 3.5.6 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.

### 3.5.7 Opérande TITRE

◇ TITRE

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



## 4 Exemples

### 4.1 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. , ),
                   )
```

### 4.2 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 : *18/02/05*  
 Clé : *U7.02.01-E* Page : *10/14*

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

1 4 3 0 2 6
1 1 1 1 1 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 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 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 10:** The value 35 is circled and labeled **POSI\_ORDRE (10,7)**.
- Record 12:** The value 8.00000E-01 is circled and labeled **POSI\_INST (12,1)**.

## 4.4 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_FICHIER = '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
```

**4.5 Exemple 5 : lecture d'un evol\_ther au format MED**

```
LIRE_RESU ( FORMAT = 'MED', MAILLAGE = MA,
             UNITE = 21, TOUT_ORDRE = 'OUI', TYPE_RESU = 'EVOL_THER',
             FORMAT_MED = _F (
                 NOM_CHAM = TEMP,
                 NOM_CHAM_MED = 'THERDEP_TEMP',
                 NOM_CMP_IDEM = 'OUI' )
             )
```

## Annexe 1 : 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'

Page laissée intentionnellement blanche.