

## 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 [ § 3.2.1.2 ] )

    / FORMAT = 'IDEAS_DS58',
      ♦ UNITE = / iunit, [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_CMP = lcmp, [l_Kn]
        ◇ NOM_CMP_MED = lcmpmed, [l_Kn]
      ), ),

    # Quelle structure de données faut-il créer ? :

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

    # Si TYPE_RESU= 'EVOL_NOLI' :

    ◇ NB_VARI = nbvar,

    # Si TYPE_RESU= 'MODE_MECA' ou 'MODE_MECA_C':

    ◇ MATR_A = matr_rigi, [matr_asse_DEPL_R]
    ◇ MATR_B = matr_mass, [matr_asse_DEPL_R]

    # autres informations :

    ♦ / MAILLAGE = ma, [maillage]

```

```

/   MODELE           =   mo,                [modele]
◇   CHAM_MATER       =   chmat,             [cham_mater]
◇   AFFE_CARA_ELEM   =   carac,             [cara_elem]

# 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-06,             [DEFAULT]
|   CRITERE          =   /'RELATIF',         [DEFAULT]
/ 'ABSOLU'

# Si TYPE_RESU= 'EVOL_THER' :

◇   EXCIT =_F(
  ◆   CHARGE          =   char,             [char_ther]
  ◇   FONC_MULT        =   fonc,             [fonction]
),

# Si TYPE_RESU= 'EVOL_NOLI' :

◇   EXCIT =_F(
  ◆   CHARGE          =   char,             [char_meca]
  ◇   TYPE_CHARGE      =   / 'FIXE_CSTE'
/ 'FIXE_PILO'
/ 'FIXE_PILO'
/ 'SUIV'
/ 'DIDI'

  ◆   /   FONC_MULT    =   fonc,             [fonction]
/   ◆   DEPL          =   depl,             [fonction]
      ◆   VITE         =   vite,             [fonction]
      ◆   ACCE         =   acce,             [fonction]
  ◇   MULT_APPUI       =   / 'OUI',
/ 'NON'             [DEFAULT]
  ◇   DIRECTION        =   (d1, d2, d3),     [l_R]
  ◇   /   NOEUD        =   lno,             [l_noeud]
/   GROUP_NO          =   lgrno,            [l_gr_noeud]
),

# Si TYPE_RESU= 'EVOL_ELAS' :

◇   EXCIT =_F(
  ◆   CHARGE          =   char,             [char_meca]
  ◇   FONC_MULT        =   fonc,             [fonction]
  ◇   TYPE_CHARGE      =   / typc           [l_Kn]
/ 'FIXE'
),

◇   TITRE              =   l_titre,          [l_Kn]
◇   INFO               =   /1,
/2
)

```

## 3 Opérandes

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

/ 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\_FICHER

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_FICHER [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.

#### Remarques :

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

*Dans certains cas particuliers (tests de non-régression) le maillage associé au modèle n'a pas été créé par `PRE_IDEAS`. Si on ne retrouve pas dans le maillage le noeud associé au numéro qui figure dans le dataset 55, alors, on récupère le nom du noeud en commentaire dans le fichier universel, tout en vérifiant que le numéro associé correspond bien au nom récupéré.*

#### 3.2.1.2 Syntaxe

```

FORMAT_IDEAS      =  _F(
♦   NOM_CHAM      =  nomch,                [1_ 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_NUME_MODE =  pnm,                  [1_I]
◇   POSI_MASS_GENE =  pmg,                  [1_I]
◇   POSI_AMOR_GENE =  pag,                  [1_I]
◇   /   POSI_INST  =  pi,                   [1_I]
◇   /   POSI_FREQ  =  pf,                   [1_I]
♦   NOM_CMP       =  lcmp,                  [1_ Kn]
◇   NB_VARI       =  nbvari,                [I]
◇   NUME_DDL      =  numddl,                [nume_ddl]
),

```

### 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 :

- : valeurs aux nœuds,
- : valeurs aux nœuds par élément
- : 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_NUME_MODE`

Vecteur de deux entiers permettant de localiser le numéro de mode

V(1) : N° de l'enregistrement

V(2) : Position du numéro de mode

◇ / `POSI_MASS_GENE`

Vecteur de deux entiers permettant de localiser la masse généralisée

V(1) : N° de l'enregistrement

V(2) : Position de la masse généralisée

◇ / `POSI_AMOR_GENE`

Vecteur de deux entiers permettant de localiser l'amortissement généralisé

V(1) : N° de l'enregistrement

V(2) : Position de l'amortissement généralisé

◇ / `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.*

## Remarque sur l'affectation du champ au nœud dans les datasets 55 :

*Dans un fichier au format universel (Ideas), un nœud du maillage est connu par son numéro. Dans Aster, un nœud est connu par son nom. La retranscription se fait en affectant au nœud, un nom qui commence par "N" suivi de son numéro Ideas. Lors de la lecture du champ au nœud dans un dataset 55, on vérifie que le nœud existe dans le maillage Aster. Si tel n'est pas le cas, alors on considère le nom donné, habituellement, en commentaire à côté du numéro du nœud dans le dataset 55.*

◇ NBVARI

Nombre de variables internes pour les champs de type ' VARI '

◇ NUME\_DDL = nume

Offre à l'utilisateur le choix de spécifier une numérotation (concept NUME\_DDL produit par la commande éponyme) pour des lectures de résultats de type `MODE_MECA`.

◇ PROL\_ZERO =

Si `PROL_ZERO` = 'OUI' : on met les valeurs des composantes aux noeuds à 0 où le champ n'est pas défini, et on émet une information du type: " Les valeurs non existantes du champ TEMP lues sur le maillage donné sont considérées nulles."

Si `PROL_ZERO` = 'NON' : on garde le champ tel quel.

Ce mot-clé est dédié uniquement aux champs aux noeuds, car pour les champs aux éléments, le champ est prolongé par 0 par défaut.

## 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) `E_DIR`=

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 associé au vecteur directeur `DIRECTION`.

## 3.4 Opérandes si `FORMAT` = 'MED'

Au format MED il n'est pas possible de lire de structures de données de type suivant : `DYNA_TRAN`, `DYNA_HARMO`, `HARM_GENE`, `MODE_MECA`, `MODE_MECA_C`. En effet, en dynamique les fichiers au format 'MED' ne permettent pas de stocker les paramètres (autres que 'INST') nécessaires aux différentes commandes.

### 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`), les champs constants (`ELEM`) et des champs aux points d'intégration (`ELGA`). 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 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.

Si ces 2 listes sont absentes, on suppose que les noms des CMPS sont les mêmes pour MED et pour Aster.

### 3.4.4 Opérande `PROL_ZERO`

Lors de la lecture d'un champ aux points de Gauss au format MED, on vérifie la cohérence entre les points de Gauss tels qu'ils sont définis dans le fichier MED et dans Aster.

Si le nombre de points de Gauss "MED" est différent du nombre de points de Gauss "Aster" on émet simplement une alarme et on ne remplit pas le champ sur les éléments en cause. On aura donc des valeurs non définies (NaN) là où le champ n'est pas défini. Le mot-clé `PROL_ZERO` = 'OUI' permet à l'utilisateur mettre des valeurs nulles s'il le désire.

## 3.5 Autres opérandes

### 3.5.1 Opérandes `TYPE_RESU/NOM_CHAM`



## ◆ TYPE\_RESU

Type de la structure de données `resultat` créée.

Pour le type `EVOL_VARC` et pour le format `MED`, si le nom du champ est `'IRRA'` alors `NOMGD='IRRA_R'`.

## ◆ NOM\_CHAM = l\_nomch

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.

Pour le format `'MED'`, le choix de la famille de points de Gauss se fait à partir du nom fourni par l'utilisateur, par exemple :

```
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 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/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 :

- 1) le `mode_meca` produit référence les 2 `matr_asse` (rigidité et masse) qui ont servi à le calculer,
- 2) 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 CHAM\_MATER/CARA\_ELEM/EXCIT

Pour qu'un résultat issu de la commande `LIRE_RESU` soit exploitable avec Stanley, il est nécessaire de disposer des informations relatives au champ de matériau et aux caractéristiques élémentaires. Par souci de complétude, les informations concernant le chargement peuvent aussi être stockées dans la structure de données résultat.

### 3.5.6 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].

### 3.5.7 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.

### 3.5.8 A propos d'une alarme

Il peut arriver que les valeurs lues sur le fichier ne puissent être recopiées dans les champs de la `SD_RESULTAT` produite. Par exemple, un champ de pression existant sur des mailles surfaciques, ne peut être recopié sur un modèle ne contenant que des éléments 3D. Dans ce cas, le code émet un message d'alarme ressemblant à :

```
<A> <LIRE_RESU> <LRCEME>
      VALEURS NON AFFECTEES DANS LE CHAMP : 3699
      VALEURS LUES DANS LE FICHIER       : 3699
```

### 3.5.9 Opérande TITRE

◇ TITRE

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

### 3.5.10 Opérande INFO

◇ INFO = inf

Permet d'imprimer dans le fichier `MESSAGE` des informations liées aux déroulement de l'opérateur.

## 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') ) ) ,
                  )
```

**Fichiers IDEAS à lire**

```

-1
57 %VALEURS AUX NOEUDS DES ELEMENTS
ASTER 3.05.30 CONCEPT U CAGUE IE - CHAMP PAR ELEMENT AUX NOEUDS DE NOM
CHAMP PAR ELEMENT AUX NOEUDS DE NOM SYMBOLE VAR1 FINO FIGA - VAR1_1 (FINO)
ASTER 3.05.30 CONCEPT U CAGUE IE 29/12/95 A 09:56:55 DE TYPE EVOL_NOLI
CHAMP PAR ELEMENT AUX NOEUDS DE NOM SYMBOLE VAR1 FINO FIGA

Record 6 →      1      4      3      0      2      6
Record 7 →      2      1      1      1
Record 8 → 0.15000E+02  ← POSI_INST (8,1)  ← POSI_ORDRE (7,4)
      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
-1
-1
57 %VALEURS AUX NOEUDS DES ELEMENTS
ASTER 3.05.30 CONCEPT U CAGUE IE - CHAMP PAR ELEMENT AUX NOEUDS DE NOM
CHAMP PAR ELEMENT AUX NOEUDS DE NOM SYMBOLE EPX EPY EPZ EPX EPY EPZ (FINO)
ASTER 3.05.30 CONCEPT U CAGUE IE 29/12/95 A 09:56:55 DE TYPE EVOL_NOLI
CHAMP PAR ELEMENT AUX NOEUDS DE NOM SYMBOLE EPSA FINO

      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
-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' ,
                                         NOME_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      35      0
Record 11 →      1500      0      POSI_ORDRE (10,7)
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  POSI_INST (12,1)
      -----
      205
      1.00000E+02
      -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',  )
           )
```

## 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'