
Sélection d'un ou plusieurs champs dans une Structure de Données RESULTAT

1 But

Description des mots-clés d'accès aux champs de grandeurs d'une structure de données de type `resultat`.

Pour extraire un champ de grandeurs dans un concept `resultat` on dispose de deux types d'indicateurs : les noms symboliques et les variables d'accès.

Ce document décrit l'ensemble des mots-clés permettant d'accéder aux champs de grandeur contenus dans une structure de données `resultat` et utilisés dans les commandes `CALC_CHAMP` [U4.81.04], `CALC_G` [U4.82.03], `COMB_SISM_MODAL` [U4.84.01], `EXTR_MODE` [U4.52.12], `IMPR_GENE` [U4.91.02], `IMPR_RESU` [U4.91.01], `LIRE_RESU` [U7.02.01], `POST_ELEM` [U4.81.22], `POST_RCCM` [U4.83.11], `POST_RELEVE_T` [U4.81.21], `RECU_FONCTION` [U4.32.03], `TEST_RESU` [U4.92.01].

2 Syntaxe

```
# Extraction d'un champ de grandeur de resu

    ◇ /      TOUT_CHAM      = / 'OUI',                [DEFAULT]
      /      / 'NON',
      /      NOM_CHAM      = L_NOMSYMB,                [1_K16]
    ◇ /      TOUT_ORDRE     = 'OUI',                [DEFAULT]
      / /      NUME_ORDRE   = LORDRE,                [1_I]
      / /      LIST_ORDRE   = LENTI,                 [listis]
      /      TOUT_MODE     = 'OUI',
      / /      NUME_MODE     = LMODE,                [1_I]
      / /      LIST_MODE     = LENTI,                 [listis]
      /      NOEUD_CMP     = LNOECMP,                [1_K16]
      /      NOM_CAS       = NCAS,                   [K16]
      /      ANGLE         = LANGL,                  [1_R]
      / /      FREQ         = LFREQ,                 [1_R]
      / /      LIST_FREQ    = LREEL,                 [listr8]
      / /      TOUT_INST    = 'OUI',
      /      INST         = LINST,                   [1_R]
      /      LIST_INST     = LREEL,                 [listr8]
    ◇ /      CRITERE       = 'RELATIF'                [DEFAULT]
      ◇ PRECISION = / prec                            [R]
      / 1.0E-3 ou 1.0D-6,*                             [DEFAULT]
      / CRITERE       = 'ABSOLU'
      ◆ PRECISION = prec                               [R]
```

* Cette valeur par défaut dépend de la structure de données de type résultat qui est exploitée.

- $PREC = 10^{-3}$ pour les commandes suivantes : TEST_RESU, EXTR_MODE, COMB_SISM_MODAL,
- $PREC = 10^{-6}$ pour les commandes suivantes : CALC_CHAMP, CALC_G, IMPR_GENE, IMPR_RESU, LIRE_RESU, POST_ELEM, POST_RCCM, POST_RELEVE_T, POST_RCCM.

La valeur par défaut de $1.0D-6$ pour la précision relative a été choisie très petite pour éviter de sélectionner des valeurs très proches obtenues par exemple lors du re-découpage automatique du pas de temps.

Le mot-clé PRECISION devient obligatoire (pas de valeur par défaut) si CRITERE='ABSOLU'.

3 Extraction d'un champ de grandeur

3.1 Choix des noms symboliques : Opérandes TOUT_CHAM / NOM_CHAM

```
/ TOUT_CHAM = /'OUI' [DEFAULT]
```

Ce mot clé indique que l'on veut accéder au(x) champ(s) de grandeurs effectivement calculé(s) pour le concept résultat RESU. Si l'on n'utilise pas l'opérande NOM_CHAM, cet opérande est pris par défaut à 'OUI'.

Les champs possibles sont décrits dans les documentations spécifiques des concepts resultat.

Exemple : les champs possibles sont pour un résultat du type evol_elas [U4.51.01] :

```
DEPL  
SIGM_ELNO  
EPSI_ELNO  
... ..
```

```
/ NOM_CHAM = L_NOMSYMB [1_K16]
```

Un concept resultat peut se composer de plusieurs champs de grandeurs de type champ aux nœuds ou champ par élément. Ce mot clé permet de choisir une liste de noms symboliques de champ parmi l'ensemble des possibles (confer TOUT_CHAM).

Remarque :

Les mots clés TOUT_CHAM et NOM_CHAM ne peuvent être utilisés simultanément.

3.2 Choix des numéros d'ordre : Opérandes TOUT_ORDRE/NUME_ORDRE/LIST_ORDRE

```
/ TOUT_ORDRE = 'OUI' [DEFAULT]
```

Ce mot clé permet d'accéder au(x) champ(s) pour tous les numéros d'ordre déjà calculés.

Exemple :

- tous les instants pour un résultat de type evol_...
- tous les modes propres pour un résultat de type mode_meca.

```
/ NUME_ORDRE : LORDRE [1_I]
```

Ce mot clé permet de n'accéder qu'au(x) champ(s) correspondant à une liste de numéros d'ordre LORDRE (le pluriel est employé ici pour le cas où l'on a choisi TOUT_CHAM = 'OUI').

Exemple : RESU = (_F(RESULTAT= RESU, TOUT_CHAM = 'OUI', NUME_ORDRE= (3,6))

On accédera à tous les champs calculés au troisième et au sixième numéro d'ordre.

```
/ LIST_ORDRE : LENTI [listis]
```

Le mot clé LIST_ORDRE indique que l'on veut accéder au(x) champ(s) correspondant à une liste de numéros d'ordre LENTI, définie par l'opérateur DEFI_LIST_ENTI [U4.34.02] (LENTI est donc un concept de type listis).

Exemple :

```
LISTE = DEFI_LIST_ENTI ( VALE = (3,6) )  
RESU = (_F ( RESULTAT= RESU, TOUT_CHAM = 'OUI', LIST_ORDRE= LISTE )
```

On accédera à tous les champs calculés au troisième et au sixième numéro d'ordre.

3.3 Autres façons de choisir les numéros d'ordre : variables d'accès

3.3.1 Opérandes TOUT_MODE/NUME_MODE/LIST_MODE/NOEUD_CMP

```
/ TOUT_MODE = 'OUI'
```

Ce mot clé indique que l'on veut accéder à tous les champs pour tous les numéros de mode déjà calculés.

Il n'est licite que pour les types de concept `resultat` (mode statiques ou dynamiques) ayant pour variable d'accès `NUME_MODE`.

```
/ NUME_MODE = LMODE [1_I]
```

Ce mot clé indique que l'on veut accéder au(x) champ(s) correspondant à une liste de numéros de mode `LMODE`. Ces numéros sont ceux affectés par l'algorithme de calcul.

Il n'est licite que pour les types de concept `resultat` (mode statiques ou dynamiques) ayant pour variable d'accès `NUME_MODE`.

Exemple : `RESU = (_F (RESULTAT= RESU, TOUT_CHAM = 'OUI', NUME_MODE= (3,6))`

On accédera à tous les champs correspondant au troisième et au sixième mode.

```
/ LIST_MODE = LENTI [listis]
```

Le mot clé `LIST_MODE` indique que l'on veut accéder au(x) champ(s) correspondant à une liste de numéros de modes `LENTI`, définie par l'opérateur `DEFI_LIST_ENTI` [U4.34.02] (`LENTI` est donc un concept de type `listis`). Il n'est valide que pour les types de concept `resultat` ayant pour variable d'accès `NUME_MODE`.

Exemple :

```
LISTE = DEFI_LIST_ENTI ( VALE = (3,6))  
RESU = (_F ( RESULTAT= RESU, TOUT_CHAM = 'OUI', LIST_MODE= LISTE )
```

On accédera à tous les champs correspondant au troisième et au sixième mode.

```
/ NOEUD_CMP = LNOECMP [1_K16]
```

Les résultats de type `base_modale` ou `mode_stat` ont pour variable d'accès `NOEUD_CMP`. La valeur de ces variables d'accès s'obtient en concaténant le nom du nœud et le nom de la composante. Pour spécifier une valeur de la variable d'accès, l'utilisateur donnera deux noms, le nom du nœud suivi du nom de la composante. Une liste de n variables d'accès sera introduite par la donnée de n couples (nom du nœud, nom de la composante).

Une liste de valeurs de cette variable d'accès peut être fournie par la liste (`couple1, couple2, ...`).

3.3.2 Opérande NOM_CAS

```
/ NOM_CAS = NCAS [K16]
```

Le mot clé `NOM_CAS` indique que l'on veut accéder au champ correspondant à un cas de chargement. Il n'est licite que pour les concepts `resultat` de type `mult_elas` qui ont pour variable d'accès `NOM_CAS` et produits par `MACRO_ELAS_MULT` [U4.51.02].

3.3.3 Opérande ANGLE

```
/ ANGLE = LANGL [1_R]
```

Le mot clé `ANGLE` indique que l'on veut accéder au(x) champ(s) recombinaison(s) pour une liste d'angles. Il n'est licite que pour les concepts `resultat` de type `comb_fourier` qui ont pour variable d'accès `ANGLE` et produits par l'opérateur `COMB_FOURIER` [U4.83.31].

3.3.4 Opérandes `FREQ` / `LIST_FREQ`

Ces opérandes ne sont licites que pour les types de concept résultat ayant pour variable d'accès `FREQ` (`mode_meca`, `dyna_harm`, ...).

```
/ FREQ = LFREQ [l_R]
```

Le mot clé `FREQ` indique que l'on veut accéder au(x) champ(s) correspondant à une liste de fréquences `LFREQ`.

Exemple : `RESU = (_F (RESULTAT= RESU, TOUT_CHAM = 'OUI', FREQ= (3.52,7.37))`

On accédera à tous les champs correspondant aux fréquences 3.52 et 7.37 .

```
/ LIST_FREQ = LREEL [listr8]
```

Le mot clé `LIST_FREQ` indique que l'on veut accéder au(x) champ(s) correspondant à une liste de fréquences `LREEL`, définie par l'opérateur `DEFI_LIST_REEL` (`LREEL` est donc un concept de type `listr8`).

Exemple :

```
LISTE = DEFI_LIST_REEL ( VALE = (3.52,7.37))  
RESU = (_F (RESULTAT= RESU, TOUT_CHAM = 'OUI', LIST_FREQ = LISTE )
```

On accédera à tous les champs correspondant aux fréquences 3.52 et 7.37 .

3.3.5 Opérandes `TOUT_INST` / `INST` / `LIST_INST`

Ces opérandes ne sont licites que pour les types de concept résultat d'évolution temporelle ayant pour variables d'accès `INST` (`evol_noli`, `evol_ther`, `dyna_trans`, ...).

```
/ TOUT_INST = 'OUI'
```

Ce mot clé indique que l'on veut accéder à tous les champs pour tous les instants déjà calculés.

```
/ INST = LINST [l_R]
```

Le mot clé `INST` indique que l'on veut accéder au(x) champ(s) correspondant à une liste d'instants `LIST_INST`.

Exemple : `RESU = (_F (RESULTAT= RESU, TOUT_CHAM = 'OUI', INST= (3. , 7.))`

On accédera à tous les champs correspondant aux instants 3. et 7. .

```
/ LIST_INST = LREEL [listr8]
```

Le mot clé `LIST_INST` indique que l'on veut accéder au(x) champ(s) correspondant à une liste d'instants `LREEL`, définie par l'opérateur `DEFI_LIST_REEL` (`LREEL` est donc un concept de type `listr8`).

Exemple :

```
LISTE = DEFI_LIST_REEL ( VALE = (3. , 7.))  
RESU = (_F ( RESULTAT= RESU, TOUT_CHAM = 'OUI', LIST_INST= LISTE )
```

On accédera à tous les champs correspondant aux instants 3. et 7. .

3.3.6 Opérandes PRECISION / CRITERE

Ces opérandes permettent d'affiner l'accès par variables d'accès réelles du temps ou de la fréquence.

```
PRECISION = / prec [R]
            / 1.0D-3 ou 1.0D-6 [DEFAULT]
```

Ce mot clé permet d'indiquer que l'on recherche tous les champs dont l'instant (respectivement la fréquence) se trouve dans l'intervalle " $inst \pm prec$ " (confer CRITERE).

Par défaut $prec = 1.0D-3$ ou $prec = 1.0D-6$ (suivant les commandes).

Si CRITERE='ABSOLU', il n'y a pas de valeur par défaut.

```
CRITERE = / 'RELATIF' [DEFAULT]
          / 'ABSOLU'
```

'RELATIF' : l'intervalle de recherche est : $[inst (1 - prec), inst (1 + prec)]$

'ABSOLU' : l'intervalle de recherche est : $[inst - prec, inst + prec]$.