

Opérateur RECU_FONCTION

1 But

Extraire sous forme d'une fonction l'évolution d'une grandeur en fonction d'une autre.

Si l'extraction s'effectue à partir d'une structure de données `resultat`, ou d'un champ de grandeur `cham_gd`, ou d'un `resu_gene`, la fonction produite correspond à l'évolution temporelle d'une composante en un nœud ou en un point de Gauss du maillage.

D'une structure de données `tran_gene`, on peut également extraire l'évolution de deux paramètres en un nœud de choc.

D'une `table`, on peut extraire l'évolution de 2 paramètres dans les colonnes de la table ou une fonction contenue dans une case de la table.

D'une `nappe`, on peut extraire la fonction correspondant à une valeur donnée du paramètre.

D'une structure de données `melasflu` on peut extraire l'évolution de paramètres modaux en fonction de la vitesse d'écoulement du fluide ou l'évolution de la vitesse critique en fonction du nombre de Connors.

D'une structure de données `interspectre` on peut extraire, l'évolution fréquentielle de l'interspectre associé à la i -ième ligne j -ième colonne de la matrice interspectrale ou l'évolution temporelle de l'interspectre à la n -ième composante de la diagonale.

Produit une structure de données de type `fonction` ou `fonction_c`.

En sortie de la commande, la fonction est réordonnée par abscisses croissantes. En revanche, il est interdit d'avoir plusieurs abscisses identiques (ce ne serait plus une fonction).

2 Syntaxe

```
fr [fonction]= RECU_FONCTION (
    ♦ / RESULTAT = resu,                                / [dyna_harmo]
                                                    / [evol_elas]
                                                    / [dyna_trans]
                                                    / [evol_ther]
                                                    / [evol_noli]

    # Voir extraction et localisation du champ
    / CHAM_GD = ch_gd,                                / [cham_no]
                                                    / [cham_elem]

    # Voir opérandes de localisation du champ
    / RESU_GENE = gene,                                / [tran_gene]

    # Évolution temporelle d'une composante physique
    ♦ / NOM_CHAM = /'DEPL',
                                                    /'VITE',
                                                    /'ACCE',
                                                    /'PTM',
    ♦ NOM_CMP = cmp,                                    [K]
    / GROUP_NO = grno,                                  [gr_noeud]
    / NUME_CMP_GENE = val_n                             [I]
    ◇ / MULT_APPUI = 'OUI',
    / | CORR_STAT = 'OUI',
    | ACCE_MONO_APPUI = frap, [fonction]
    / GROUP_NO_CHOC= no_choc, [gr_noeud]
    ♦ PARA_X = nparax, [Kn]
    ♦ PARA_Y = nparay, [Kn]
    ◇ INTITULE = nom, [Kn]
    ◇ LIST_PARA = li_para, [listr8]
    ◇ SOUS_STRUC = nom_str, [Kn]
    / RESU_GENE = gene, [harm_gene]

    # Évolution fréquentielle d'une composante généralisée ou physique
    ♦ NOM_CHAM = nomsymb, [K16]
    ♦ / NUME_CMP_GENE = numcmp, [K8]
    / NOM_CMP = cmp, [K]
    ♦ GROUP_NO = grno, [gr_noeud]
    / RESU_GENE = gene, [mode_gene]

    # Évolution fréquentielle d'une composante généralisée ou physique
    ♦ / NOM_PARA_RESU = parametre, [K8]
    / NOM_CHAM = nomsymb, [K16]
    ♦ / NUME_CMP_GENE = numcmp, [K8]
    / NOM_CMP = cmp, [K]
    ♦ GROUP_NO = grno, [gr_noeud]
    ♦ / SQUELETTE = squ, [squelette]
    / SOUS_STRUC = sstru, [K]
```

```

~  TABLE =      tabl,                                /  [table]
      ♦ PARA_X   =  nparax,                            [Kn]
      ♦ PARA_Y   =  nparay,                            [Kn]
      ◇ NOM_PARA_TABL = / 'FONCTION',                [Kn]
                                / 'FONCTION_C'

      ◇ FILTRE = _F(
          ♦ NOM_PARA =                                [Kn]
          ◇ CRIT_COMP= / 'EQ',                        [DEFAULT]
                                / 'LT',
                                / 'GT',
                                / 'NE',
                                / 'LE',
                                / 'GE',
                                / 'VIDE',
                                / 'NON_VIDE',
                                / 'MAXI',
                                / 'MAXI_ABS',
                                / 'MINI',
                                / 'MINI_ABS',

          ♦ / VALE   = val_r,                        [R]
            / VALE_I = val_n,                        [I]
            / VALE_C = val_c,                        [C]
            / VALE_K = val_k,                        [Kn]
          ◇ | CRITERE = / 'RELATIF',                [DEFAULT]
                                / 'ABSOLU',
            | PRECISION = / prec,
                                / 0.001,            [DEFAULT]
          ),

/  BASE_ELAS_FLUI = flui,                            [melasflu]
      ♦ NUME_MODE = im,                              [I]
      ♦ / PARA_X   = / 'VITE_FLU',                  [Kn]
          ♦ / TOUT_ORDRE = 'OUI',                    [DEFAULT]
          / NUME_ORDRE = is,                          [I]
          ♦ PARA_Y   = / 'FREQ',                    [Kn]
                                / 'AMOR',

      / PARA_X   = / 'NB_CONNORS',                  [Kn]
      ♦ PARA_Y   = / 'VITE_CRIT',                    [Kn]
                                / 'INSTAB_EDF',
                                / 'INSTAB_TOUT_CMP',

/  INTE_SPEC = intespec,                            [interspectre]
# Évolution fréquentielle d'une composante de la matrice interspectrale
      ◇ NOM_CHAM   = nomsymb,                        [K16]
      ♦ / ♦ NUME_ORDRE_I = numei,                    [I]
          ◇ NUME_ORDRE_J = numej,                    [I]
          / ♦ NUME_ORDRE = numei,                    [I]
          / ♦ NOEUD_I   = noei,                      [noeud]
          ♦ NOM_CMP_I   = cmpi,                      [Kn]
          ◇ NOEUD_J   = noej,                      [noeud]
          ◇ NOM_CMP_J   = cmpj,                      [Kn]

/  NAPPE = nap,                                      [nappe]
      ♦ VALE_PARA_FONC = np,                        [Kn]
      ◇ | CRITERE = / 'RELATIF',                    [DEFAULT]
                                / 'ABSOLU',
            | PRECISION = / prec,
                                / 0.001,            [DEFAULT]

# Opérandes d'extraction du champ ou du paramètre si on

```

```
manipule une SD_resultat ou RESU_GENE ou un cham_gd

    ♦ / NOM_CHAM      = nomsymb,           [K16]
      / NOM_PARA_RESU = parametre,
    ◇ / TOUT_ORDRE    = 'OUI',             [DEFAULT]
      / TOUT_INST     = 'OUI',
      / NUME_ORDRE    = l_nume,             [l_I]
      / LIST_ORDRE    = l_ord,              [listis]
      / / INST        = l_inst,             [l_R]
      / LIST_INST     = li_inst,            [listr8]
      / FREQ          = l_freq,             [l_R]
      / LIST_FREQ     = li_freq,            [listr8]
    ◇ | PRECISION     = / prec,             [R]
      |               / 1.0D-3,            [DEFAULT]
      | CRITERE       = / 'RELATIF',        [DEFAULT]
      |               / 'ABSOLU',
    ◇ INTERP_NUME     = / 'NON',            [DEFAULT]
      |               / 'LIN',

# Opérandes de localisation du champ si on manipule un resultat ou un cham_gd

    ♦ / GROUP_NO      = grno,              [gr_noeud]
      / GROUP_MA      = grma,              [gr_maille]
      | POINT         = nupoint,           [I]
      | SOUS_POINT    = nusp,              [I]
    ♦ NOM_CMP         = cmp,               [K]

# Surcharge des attributs de la fonction créée

    ◇ NOM_PARA        = nom_pa,            [Kn]
    ◇ NOM_RESU        = nom_res,           [Kn]
    ◇ INTERPOL        = / | 'LIN',          [Kn]
      |               | 'LOG',
    ◇ PROL_DROITE     = / 'CONSTANT',
      |               / 'LINEAIRE',
      |               / 'EXCLU',
    ◇ PROL_GAUCHE     = / 'CONSTANT',
      |               / 'LINEAIRE',
      |               / 'EXCLU',

    ◇ TITRE          = t,                  [l_K]
    ◇ INFO           = / 1,                [DEFAULT]
      |               / 2,

)

Si RESULTAT est un [dyna_harmo] alors fr est [fonction_c],
Si RESU_GENE est un [harm_gene] alors fr est [fonction_c],
Si INTE_SPEC et NUME_ORDRE_J ou NOEU_J alors fr est [fonction_c],
Si NOM_PARA_TABL = 'FONCTION_C' alors fr est [fonction_c],
Dans les autres cas, fr est [fonction].
```

3 Opérandes

3.1 Opérande **RESULTAT**

- ♦ `RESULTAT = resu`
Nom du concept de type `resultat` sur lequel porte l'extraction.
Pour les opérandes permettant d'extraire le champ, se reporter au [§3.7].
Pour les opérandes permettant de localiser le champ, se reporter au [§3.8].

3.2 Opérande **CHAM_GD**

- ♦ `CHAM_GD = ch_gd`
Nom du concept d'un champ sur lequel porte l'extraction. Pour les opérandes permettant de localiser le champ, se reporter au [§3.7].

Le champ fourni au mot clé `CHAM_GD` est :
 - soit un champ aux nœuds de grandeur : `DEPL_R`, `TEMP_R` ou `PRES_R` ;
 - soit un champ par éléments (aux nœuds ou aux points de Gauss) de grandeur :
`VARI_R`, `EPSI_R`, `FLUX_R`, ou `PRES_R`.

3.3 Opérande **RESU_GENE**

3.3.1 Évolution temporelle d'une composante physique ou généralisée, type **tran_gene**

- ♦ `RESU_GENE = gene`
Nom du concept de type `resu_gene` produit par `DYNA_TRAN_MODAL` [U4.53.21] sur lequel porte l'extraction.

La fonction récupérée est exprimée
 - selon les variables physiques : on a précisé `GROUP_NO = grno`.
 - selon les variables généralisées : on a précisé `NUME_CMP_GENE = n_val`.
`NOM_CHAM` donne le nom du champ que l'on veut récupérer ('`DEPL`', '`VITE`', '`ACCE`', ou '`PTEM`'). L'option '`PTEM`' permet d'extraire, pour chaque instant (ou numéro d'ordre) de récupération, les valeurs du pas de temps de calcul.

Remarque :

Faire cette restitution sur base physique est le rôle de la commande `REST_GENE_PHYS`. Dans les versions ultérieures, cette fonctionnalité sera retirée de `RECU_FONCTION`, il faudra faire la restitution puis extraire la fonction.

3.3.1.1 Opérandes **MULT_APPUI** et **ACCE_MONO_APPUI**

- ◇ `MULT_APPUI`
Si ce mot clé est '`OUI`', on restitue l'évolution des variables dans l'espace physique en traitant le problème en mouvement absolu dans le cas d'une excitation multi-appui. Dans le cas contraire, la restitution dans l'espace physique se fait en supposant que le problème est traité en mouvement relatif. Ce mot-clé n'est pas utilisable si le mot-clé `CORR_STAT` est utilisé.
- ◇ `ACCE_MONO_APPUI`
Dans le cas d'une accélération mono-appui, on doit indiquer ici l'accélération imposée à tous les supports dans la direction considérée afin de calculer l'accélération absolue du point.

Si le mot clé n'est pas renseigné, on obtient l'accélération relative en résultat de la commande.

Remarque :

Le nom du concept doit être le même que celui renseigné sous `FONC_MULT` de `DYNA_TRAN_MODAL`.

3.3.1.2 Opérandes CORR_STAT◇ `CORR_STAT`

Si ce mot clé est 'OUI', l'évolution des variables dans l'espace physique est obtenue en tenant compte de la correction due à la prise en considération de modes statiques (Cf. [R4.05.03]). Ce mot-clé n'est pas utilisable si le mot-clé `MULT_APPUI` est utilisé.

3.3.1.3 Informations concernant les nœuds de choc◆ `RESU_GENE = gene`

Concept de type `tran_gene` contenant pour les différents nœuds de choc : les déplacements locaux, les vitesses normales et tangentielles et les forces de choc normales et tangentielles.

◆ `GROUP_NO_CHOC = no_choc,`

Groupe de nœuds (qui ne contient qu'un seul nœud) de choc où on récupère la fonction.

Ce nœud de choc est défini dans la commande `DYNA_TRAN_MODAL` [U4.53.21].

◆ `PARA_X = nparax`

Nom du paramètre définissant les abscisses (argument pris parmi la liste : 'INST', 'FN', 'FT1', 'FT2', 'VN', 'VT1', 'VT2', 'DXLOC', 'DYLOC', 'DZLOC', 'VINT1', 'VINT2', 'VINT3', 'VINT4', 'VINT5', 'VINT6', 'VINT7', 'VINT8', 'VINT9', 'VINT10', 'VINT11', 'VINT12', 'VINT13', 'VINT14', 'VINT15', 'VINT16', 'VINT17', 'VINT18', 'VINT19', 'VINT20', 'VINT21').

◆ `PARA_Y = nparay`

Nom du paramètre définissant les ordonnées (argument pris parmi la liste : 'INST', 'FN', 'FT1', 'FT2', 'VN', 'VT1', 'VT2', 'DXLOC', 'DYLOC', 'DZLOC', 'VINT1', 'VINT2', 'VINT3', 'VINT4', 'VINT5', 'VINT6', 'VINT7', 'VINT8', 'VINT9', 'VINT10', 'VINT11', 'VINT12', 'VINT13', 'VINT14', 'VINT15', 'VINT16', 'VINT17', 'VINT18', 'VINT19', 'VINT20', 'VINT21').

◇ `LIST_PARA = li_para`

Liste des valeurs du paramètre en abscisse définissant la fonction.

Attention : Il est possible d'avoir des problèmes d'interpolation du résultat sur cette liste de paramètres (car à la précision machine près, les valeurs peuvent être légèrement hors des bornes des fonctions produites). Dans ce cas, il suffit de ne pas utiliser ce mot-clé ici. La fonction sera alors créée sur tous les instants de calcul. On peut ensuite interpoler cette fonction avec `CALC_FONC_INTERP` sur la liste des paramètres de son choix en maîtrisant les prolongements à gauche et à droite.

◇ `INTITULE = nom`

Ce nom définit la liaison de choc (ce nom s'il est utilisé, est défini dans la commande `DYNA_TRAN_MODAL` [U4.53.21]).

◇ `SOUS_STRUC = nom_str`

Lors d'un calcul en sous-structuration dynamique, nom de la sous-structure qui contient le nœud de choc (cf. commande **DEFI_MODELE_GENE** [U4.65.02]). Dans ce cas le mot clé **INTITULE** doit être aussi renseigné.

3.3.2 Évolution fréquentielle d'une composante généralisée ou physique, type **harm_gene**

♦ **RESU_GENE** = gene

Nom du concept de type **harm_gene** produit par **DYNA_LINE_HARM** [U4.53.11].

La fonction récupérée est exprimée

- selon les variables physiques : on a précisé **GROUP_NO** = grno.
- selon les variables généralisées : on a précisé **NUME_CMP_GENE** = n_val.

NOM_CHAM donne le nom du champ que l'on veut récupérer ('DEPL', 'VITE' ou 'ACCE').

Remarque :

*Faire cette restitution sur base physique est le rôle de la commande **REST_GENE_PHYS**. Dans les versions ultérieures, cette fonctionnalité sera retirée de **RECU_FONCTION**, il faudra faire la restitution puis extraire la fonction.*

3.3.3 Évolution fréquentielle d'une composante généralisée ou physique, type **mode_gene**

♦ **RESU_GENE** = gene

Nom du concept de type **mode_gene** produit par **CALC_MODES** [U4.53.02].

La fonction récupérée est exprimée avec les variables physiques si **NOM_CMP** est présent, avec les variables généralisées si **NUME_CMP_GENE** est présent.

♦ **NOM_PARA_RESU** / **NOM_CHAMP** Voir paragraphe 3.8.

♦ **NOM_CMP** / **GROUP_NO** Voir paragraphe 3.9.

♦ **SQUELETTE** Nom du maillage squelette de la structure globale sur lequel le résultat sera restitué : voir l'opérateur **DEFI_SQUELETTE** [U4.24.01].

♦ **SOUS_STRUC** Voir ci-dessus.

3.4 Opérande **TABLE**

On peut récupérer :

- 1) soit une fonction définie à partir de deux colonnes de la table,
- 2) soit une fonction dont le nom est indiquée dans une case de la table.

♦ **TABLE** = tabl Nom de la table résultat dans laquelle on effectue une extraction.

3.4.1 Fonction définie à partir de deux colonnes de la table

3.4.1.1 Opérandes **PARA_X** / **PARA_Y**

♦ **PARA_X** = nparax

Nom de la colonne de la table définissant les abscisses.

♦ **PARA_Y** = nparay

Nom de la colonne de la table définissant les ordonnées.

3.4.2 Fonction dont le nom est indiquée dans une case de la table

3.4.2.1 Opérande **NOM_PARA_TABL**

- ♦ `NOM_PARA_TABL = 'FONCTION' ou 'FONCTION_C'`

La présence de ce mot-clé indique qu'on récupère une fonction dont le nom est stocké dans une case de la table. Les fonctions réelles sont stockées dans la colonne 'FONCTION', les fonctions complexes dans la colonne 'FONCTION_C'.

3.4.2.2 Mot clé **FILTRE**

- ♦ `FILTRE`

Les opérandes d'extraction sont différents de ceux utilisés pour les cas précédents. Pour réaliser l'extraction, il faut utiliser le mot-clé `FILTRE` et les opérandes `NOM_PARA`, `CRIT_COMP`, `VALE_X`, `CRITERE`, `PRECISION`.

Ce mot-clé facteur permet de filtrer les informations stockées dans la table. Pour l'utilisation de ce mot-clé voir la commande `IMPR_TABLE` [U4.91.03].

Pour récupérer une fonction dont le nom est indiqué dans une case de la table, Il faut utiliser au moins deux fois le mot-clé facteur `FILTRE` pour ne sélectionner que la case utile.

3.5 Opérande **BASE_ELAS_FLUI**

On récupère dans une structure de données de type `melasflu` produite par l'opérateur `CALC_FLUI_STRU` [U4.66.02], les évolutions de la fréquence ou de l'amortissement, pour un mode donné, en fonction des différentes vitesses d'excitation du fluide.

- ♦ `BASE_ELAS_FLUI = flu`

Concept de type `melasflu` produit par la commande `CALC_FLUI_STRU`.

3.5.1 Opérande **NUME_MODE**

- ♦ `NUME_MODE = im`

Numéro du mode pour lequel l'extraction de la fréquence ou de l'amortissement en fonction de la vitesse du fluide est effectuée.

3.5.2 Opérandes **PARA_X / PARA_Y/NUME_ORDRE/TOUT_ORDRE**

- ♦ `PARA_X = / 'VITE_FLU'`
`/ 'NB_CONNORS'`

En abscisse, le paramètre est la vitesse d'excitation du fluide, de nom 'VITE_FLU' ou bien le nombre de Connors, de nom 'NB_CONNORS'.

Si l'on a choisi comme abscisse la vitesse d'excitation du fluide 'VITE_FLU', alors :

- ♦ `PARA_Y = / 'FREQ',`
`/ 'AMOR'.`

En ordonnée, on a le choix entre la fréquence (nom du paramètre 'FREQ') ou l'amortissement (nom du paramètre 'AMOR').

- ♦ `/ NUME_ORDRE = is,`
`/ TOUT_ORDRE = 'OUI',`

L'évolution de la fréquence ou celle de l'amortissement est donnée pour toutes les vitesses du fluide (`TOUT_ORDRE`) ou pour quelques numéros d'ordre des vitesses du fluide (`NUME_ORDRE`).

Si l'on a choisi comme abscisse le nombre de Connors 'NB_CONNORS', alors il existe trois possibilités pour l'ordonnée :

- ♦ `PARA_Y = / 'VITE_CRIT', [Kn]`


```
/'INSTAB_EDF',  
/'INSTAB_TOUT_CMP',
```

VITE_CRIT permet de tracer la vitesse critique en fonction du nombre de Connors. Pour accéder aux rapports d'instabilité (vitesse efficaces divisées par la vitesse critique), les choix de PARA_Y entre INSTAB_EDF et INSTAB_TOUT_CMP permettent de tracer le rapport calculé avec la méthode Gevibus (méthode EDF) et la méthode conservative (dite toutes composantes) respectivement [R4.07.04].

3.6 Opérande INTE_SPEC

On extrait dans une structure de données de type `interspectre`, l'évolution fréquentielle de l'interspectre associé à ligne *i* et la colonne *j* de la matrice interspectrale `intespec`. On peut également extraire l'évolution temporelle d'une des fonctions générées par GENE_FONC_ALEA [U4.36.05] en spécifiant son numéro d'ordre.

♦ `INTE_SPEC = intespec`

3.6.1 Opérande NOM_CHAM

♦ `NOM_CHAM = nomsymb`

Nom symbolique du champ sur lequel porte l'extraction.

3.6.2 Opérandes NUME_ORDRE_I, NUME_ORDRE_J

♦ `NUME_ORDRE_I = numei`

♦ `NUME_ORDRE_J = numej`

Indication du couple d'indices (ligne *i*, colonne *j*) à extraire de la matrice interspectrale `intespec`.

Ces opérandes s'excluent avec les opérandes `NUME_ORDRE` `NOEUD_I` `NOM_CMP_I` `NOEUD_J` `NOM_CMP_J`.

Remarque :

Il n'est pas obligatoire de renseigner NUME_ORDRE_J si on veut extraire un terme de la diagonale de la matrice.

3.6.3 Opérandes NUME_ORDRE

♦ `NUME_ORDRE = nume`

Indication du numéro d'ordre de la fonction temporelle générée par l'opérateur GENE_FONC_ALEA [U4.36.05].

Ces opérandes s'excluent avec les opérandes `NUME_ORDRE_I` `NUME_ORDRE_J` `NOEUD_I` `NOM_CMP_I` `NOEUD_J` `NOM_CMP_J`.

3.6.1 Opérandes NOEUD_I, NOM_CMP_I, NOEUD_J, NOM_CMP_J

♦ `NOEUD_I = noei`

♦ `NOM_CMP_I = cmpi`

♦ `NOEUD_J = noej`

♦ `NOM_CMP_J = cmpj`

Ces opérandes correspondent aux noms des nœuds et des composantes (ligne *i*, colonne *j*) de la matrice de la matrice interspectrale `intespec`.

Ces opérandes s'excluent avec les opérandes NUME_ORDRE_I NUME_ORDRE_J
NUME_ORDRE.

Remarque :

Il n'est pas obligatoire de renseigner NOEUD_J et NOM_CMP_J si on veut extraire un terme de la diagonale de la matrice.

3.7 Opérande NAPPE

On récupère dans une structure de données de type `nappe` la fonction correspondant à une valeur donnée du paramètre de la nappe.

♦ `VALE_PARA_FONC = np`

`np` est la valeur du paramètre de la nappe pour laquelle on souhaite extraire la fonction.
Il n'y a pas d'interpolation sur le paramètre de la nappe. `CRITERE` et `PRECISION` permettent de fournir `np` avec une précision donnée.

3.8 Opérandes d'extraction du champ ou du paramètre

3.8.1 Opérande NOM_CHAM

♦ `NOM_CHAM = nomsymb`

Nom symbolique du champ sur lequel porte l'extraction.

3.8.2 Opérande NOM_PARA_RESU

♦ `NOM_PARA_RESU = parametre`

Nom symbolique du paramètre de la structure de données que l'on veut extraire (par exemple : `ETA_PILOTAGE`, `MASSE_EFFE_DX`, `MASSE_GENE` ...).

La fonction extraite aura alors pour abscisse la variable d'accès (`INST`, `FREQ`...) et pour ordonnée la valeur de `parametre`.

3.8.3 Opérandes TOUT_ORDRE / NUME_ORDRE / TOUT_INST / LIST_ORDRE

♦ `/ TOUT_ORDRE = 'OUI'` (valeur par défaut)

Ce mot clé indique que l'on veut extraire pour tous les numéros d'ordre déjà calculés.

Exemple : tous les instants pour un résultat de type `evol_*`.

`/ NUME_ORDRE = l_num`

L'extraction se fera pour les valeurs de numéro d'ordre `l_num` fournies.

`/ TOUT_INST = 'OUI'`

Ce mot clé indique que l'on veut extraire pour tous les instants.

`/ LIST_ORDRE = l_ord`

Ce mot clé indique que l'on veut extraire aux numéros d'ordre décrits dans le concept `l_ord` de type `listis`.

3.8.4 Opérandes INST / LIST_INST / FREQ / LIST_FREQ

♦ `/ INST = l_inst`

Ce mot clé indique que l'on veut extraire aux instants `l_inst`.

/ LIST_INST = li_inst

Ce mot clé indique que l'on veut extraire aux instants décrits dans le concept li_inst de type listr8.

/ FREQ = l_freq

Ce mot clé indique que l'on veut extraire aux fréquences l_freq.

/ LIST_FREQ = li_freq

Ce mot clé indique que l'on veut extraire aux fréquences décrites dans le concept li_freq de type listr8.

3.8.5 Opérandes PRECISION / CRITERE

◇ PRECISION = prec

Cet opérande permet d'indiquer que l'on recherche la valeur du champ dont l'instant ou la fréquence se trouve dans un intervalle défini par la position absolue ou relative : "inst ± prec" (Cf. CRITERE).

Par défaut prec = 1.0D-3

◇ CRITERE =

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

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

3.8.6 Opérande INTERP_NUME

◇ INTERP_NUME

Ce mot clé définit le type d'interpolation entre deux numéros d'ordre. Il n'est valable que dans le cas où l'utilisateur a défini une liste d'instant ou de fréquences. Il est possible d'interdire l'interpolation 'NON' ou d'admettre une interpolation linéaire 'LIN'.

L'interpolation ne peut pas être utilisée lorsqu'on extrait la valeur d'un paramètre (mot clé NOM_PARA_RESU).

3.9 Opérandes de localisation du champ

3.9.1 Opérande GROUP_NO

◆ GROUP_NO = grno

Nom du groupe de nœuds, contenant 1 seul nœud, sur lequel porte l'extraction.

3.9.2 Opérandes GROUP_MA / GROUP_NO / POINT

◆ GROUP_MA = grma

Nom d'un groupe de mailles (grma), contenant une seule maille, sur laquelle porte l'extraction. Ces mots-clés ne concernent que les cham_elem.

◆ / GROUP_NO = grno

Désigne le nom du groupe de nœuds, contenant un seul nom de nœud, sur lequel porte l'extraction (cas des cham_elem aux nœuds).

/ POINT = nupoint

L'entier `nupoint` précise le numéro local à l'élément du point de GAUSS duquel on souhaite obtenir la valeur (cas des `cham_elem` aux points de GAUSS).

◇ `SOUS_POINT = nusp`

L'entier `nusp` précise le numéro du sous-point duquel on souhaite obtenir la valeur (cas des `cham_elem` à sous-points, utilisés par les éléments de structure : poutre, tuyaux, coques).

Dans le cas des plaques et des coques multicouches, le numéro du sous-point correspond au niveau dans l'ensemble des couches. Chaque couche est décrite par une peau inférieure, moyenne et supérieure. Par convention, pour N couches, ce numéro varie entre 1 et $3N$ où le premier point se situe au niveau de la peau inférieure de la première couche et le $3N$ ième point au niveau de la peau supérieure de la dernière couche (cf. [R3.07.03] et [R3.07.04] pour la numération des couches).

Dans le cas des poutres multifibres, cet entier est le numéro de la fibre dont la numérotation est décrite dans la documentation [U4.26.01] et [R3.08.08].

Dans le cas des tuyaux, il faut se référer à la description faite dans le document [R3.08.06].

3.9.3 Opérandes `NOM_CMP/NOM_VARI`

◇ `NOM_CMP = nocmp,` [Kn]

Nom de la composante sur laquelle porte l'extraction .

◇ `NOM_VARI = novari ,` [Kn]

Pour les champs des variables internes (`VARI_*`), on peut donner le nom de la variable interne sur laquelle porte l'extraction (voir [U4.51.11] pour les règles de nommage des variables internes).

3.10 Attributs du concept fonction créé par RECU_FONCTION

3.10.1 Valeurs par défaut

Par défaut les attributs du concept fonction créé par la commande `RECU_FONCTION` sont :

Prolongement gauche : 'EXCLU'

Prolongement droit : 'EXCLU'

`NOM_PARA` : donné en entrée

`NOM_RESU` : donné en entrée

3.10.2 Surcharge des attributs

L'utilisateur peut surcharger les attributs donnés par défaut en utilisant les mots clé suivants :

3.10.2.1 Opérande `NOM_PARA`

◇ `NOM_PARA = para`

Il désigne le nom du paramètre (variable ou abscisse) de la fonction. Les valeurs actuellement autorisées pour `lpara` sont :

/ 'TEMP'	/ 'INST'	/ 'EPSI'
/ 'X'	/ 'Y'	/ 'Z'
/ 'FREQ'	/ 'PULS'	/ 'AMOR'
/ 'DX'	/ 'DY'	/ 'DZ'
/ 'DRX'	/ 'DRY'	/ 'DRZ'

plus ceux spécifiques aux nœuds de choc (cf. [§3.3.2.2]).

3.10.2.2 Opérande `NOM_RESU`

◇ `NOM_RESU = resu`

Il désigne le nom du résultat, la fonction ainsi créée est une fonction dont la valeur est de `nom_lresu` (8 caractères).

3.10.2.3 Opérande `INTERPOL`

◇ `INTERPOL`

Type d'interpolation de la fonction entre les valeurs du paramètre du domaine de définition. Derrière ce mot clé on attend une liste de paramètres (deux au maximum) parmi '`LIN`', '`LOG`'. Si une seule valeur est donnée l'interpolation sera identique pour les abscisses et les ordonnées. Si deux valeurs sont données, la première correspond à l'interpolation des abscisses et la deuxième à l'interpolation des ordonnées.

3.10.2.4 Opérandes `PROL_DROITE` / `PROL_GAUCHE`

◇ `PROL_DROITE` et `PROL_GAUCHE`

Ils définissent le type de prolongement à droite (à gauche) du domaine de définition de la variable :

- '`CONSTANT`' pour un prolongement avec la dernière (ou première) valeur de la fonction,
- '`LINEAIRE`' pour un prolongement le long du premier segment défini (`PROL_GAUCHE`) ou du dernier segment défini (`PROL_DROITE`),
- '`EXCLU`' si l'extrapolation des valeurs en dehors du domaine de définition du paramètre est interdite.

3.11 Opérande `TITRE`

◇ `TITRE`

Titre attaché au concept produit par cet opérateur [U4.03.01].

3.12 Opérande `INFO`

◇ `INFO`

Précise les options d'impression sur le fichier `MESSAGE`.

- 1 pas d'impression (par défaut)
- 2 impression du descripteur de la fonction et de la liste des 10 premières valeurs de la fonction dans l'ordre croissant des 10 premiers paramètres

4 Exemples

4.1 Extractions de fonction sur la réponse dynamique d'un réseau de tuyauterie

```
tran_gen = DYNA_TRAN_MODAL( ... )

l_inst = DEFI_LIST_REEL( DEBUT = 0.,
                        INTERVALLE = _F( JUSQU_A = 3., PAS = 0.005 ) )

dyn_tran = REST_GENE_PHYS( RESU_GENE = tran_gen, NOM_CHAM = 'DEPL',
                           LIST_INST = l_inst, INTERPOL = 'LIN' )

dyn_tran = CALC_CHAMP( ... ,
                       CONTRAINTE = 'SIEF_ELGA' )

tab_rele = POST_RELEVE_T( ACTION=_F( INTITULE = 'sixx_254',
                                     CHEMIN = ligne,
                                     RESULTAT = dyn_tran,
                                     NOM_CHAM = 'SIEF_ELGA',
                                     INST = 2.54,
                                     TOUT_CMP = 'OUI',
                                     OPERATION = 'EXTRACTION' ) )
```

4.1.1 Évolution du déplacement du nœud NO01 composante 'DX' à tous les instants de calcul

```
f1 = RECU_FONCTION( RESU_GENE = tran_gen, NOM_CHAM = 'DEPL',
                   GROUP_NO = 'GNO01' , NOM_CMP = 'DX' )
```

4.1.2 Évolution de la grandeur 'SIXX' sur la maille MA01 au nœud NO01 à tous les instants de calcul

```
f2 = RECU_FONCTION( RESULTAT= dyn_tran, NOM_CHAM= 'SIEF_ELGA',
                   GROUP_MA = 'GMA01' , GROUP_NO = 'NO01',
                   NOM_CMP='SIXX')
```

4.1.3 Évolution de la grandeur 'SIXX' le long de la ligne de tuyauterie à l'instant de calcul 2.54 s

```
f3 = RECU_FONCTION( TABLE = tab_rele,
                   PARA_X = 'ABSC_CURV', PARA_Y = 'SIXX' )
```

4.1.4 Évolution de la grandeur 'SIXX' le long de la ligne de tuyauterie (abscisse curviligne supérieure à 10) à l'instant de calcul 2.54 s

```
f4 = RECU_FONCTION( TABLE = tab_rele,
                   FILTRE = _F( NOM_PARA = 'ABSC_CURV',
                               CRIT_COMP = 'GE',
                               VALE = 10. , ),
                   PARA_X = 'ABSC_CURV', PARA_Y = 'SIXX' )
```

4.2 Extraction de fonction dans une structure de donnée melasflu

```
meles1    = CALC_FLUI_STRU ( ... )

f_freq    = RECU_FONCTION (  BASE_ELAS_FLUI = meles1,
                             PARA_X         = 'VITE_FLU',
                             PARA_Y         = 'FREQ',
                             TOUT_ORDRE     = 'OUI',
                             NUME_MODE      = 2
                             )
```

4.3 Extraction de fonction dans une structure de donnée interspectre

```
reppx_ac  = REST_SPEC_PHYS ( ... )

statx_ac  = POST_DYNA_ALEA ( INTERSPECTRE = _F( INTE_SPEC  = reppx_ac,
                                                  OPTION     = 'DIAG',
                                                  )
                          )

f_freq    = RECU_FONCTION (  INTE_SPEC      = statx_ac,
                             NOEUD_I        = 'N_TUB_01',
                             NOM_CMP_I      = 'DX',
                             )
```