Opérateur CALC_FONCTION

1 But

Effectuer des opérations mathématiques sur des structures de données de type fonction.

L'opérateur permet de réaliser des opérations générales sur les fonctions (intégration, dérivée, valeur absolue, combinaison, régression polynomiale…) et des opérations dédiées au traitement de signal et au séisme (calcul de FFT, lissage enveloppe, correction d'accélérogrammes...).

Produit une structure de données `fonction`, `fonction_c` ou `nappe`, selon le mot clé facteur utilisé.
En sortie de la commande, la fonction est réordonnée par abscisses croissantes.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

Table des matières

1 But ... 1
2 Syntaxe ... 4
3 Opérandes ... 10
3.1 Mot clé DERIVE ... 10
3.2 Mot clé INTEGRE ... 10
3.3 Mot clé INVERSE ... 11
3.4 Mot clé ABS ... 11
3.5 Mot clé ENVELOPPE ... 11
3.5.1 Opérande FONCTION ... 11
3.5.2 Opérande CRITERE ... 12
3.6 Mot clé FRACTILE ... 12
3.6.1 Opérande FONCTION ... 12
3.6.2 Opérande FRACT ... 12
3.7 Mot clé MOYENNE ... 12
3.8 Mot clé COHERENCE ... 12
3.8.1 Opérandes NAPPE_1 et NAPPE_2 ... 13
3.8.2 Opérande FREQ_COUP ... 13
3.8.3 Opérande NB_FREQ_LISS ... 13
3.8.4 Opérandes OPTION, BORNE_INF et BORNE_SUP ... 14
3.9 Mot clé COMB et opérande LIST_PARA ... 14
3.10 Mot clé COMB_C et opérande LIST_PARA ... 14
3.11 Mot clé MULT et opérande LIST_PARA ... 15
3.12 Mot clé COMPOSE ... 15
3.13 Mot clé ASSE ... 15
3.13.1 Opérande FONCTION ... 15
3.13.2 Opérande SURCHARGE ... 15
3.13.3 Vérifications ... 16
3.14 Mot clé EXTRACTION ... 16
3.14.1 Opérande FONCTION ... 16
3.14.2 Opérande PARTIE ... 16
3.15 Mot clé PUISSANCE ... 16
3.16 Mot-clé REGR_POLYNOMIALE ... 16
3.17 Mot clé FFT ... 17
3.18 Mot clé INTERPOL_FFT ... 18
3.19 Mot clé CORR_ACCE ... 18
3.19.1 Opérande FONCTION ... 18
3.19.2 Opérande METHODE ... 18
3.20 Mot clé LISS_ENVELOP ... 19

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

3.21 Mot clé SPEC_OSCI ... 20
3.21.1 Opérande FONCTION ... 21
3.21.2 Opérandes NATURE / NATURE_FONC ... 21
3.21.3 Opérande METHODE ... 22
3.21.4 Opérande TYPE_RESU ... 22
3.21.5 Opérande AMOR_REDUIT ... 22
3.21.6 Opérandes FREQ / LIST_FREQ ... 22
3.21.7 Opérande NORME ... 22
3.21.8 Opérande DUREE ... 22
3.22 Mot clé PROL_SPEC_OSCI ... 23
3.22.1 Opérande FONCTION ... 23
3.22.2 Opérande NORME ... 23
3.22.3 Opérande DEPL_MAX ... 23
3.23 Mot clé DSP ... 23
3.23.1 Opérande FONCTION ... 23
3.23.2 Opérandes AMOR_REDUIT, LIST_FREQ, FREQ_PAS ... 24
3.23.3 Opérandes FREQ_COUP ... 24
3.23.4 Opérande DUREE ... 24
3.23.5 Opérande NORME ... 24
3.23.6 Opérande FRACT ... 24
3.23.7 Opérande NB_ITER ... 24
3.24 Attributs du concept fonction en sortie ... 24
3.24.1 Valeurs par défaut ... 24
3.24.2 Surcharge des attributs ... 25
3.24.2.1 Opérande NOM_PARA ... 25
3.24.2.2 Opérande NOM_RESU ... 26
3.24.2.3 Opérande INTERPOL ... 26
3.24.2.4 Opérandes PROL_DROITE / PROL_GAUCHE ... 26
3.24.2.5 Opérandes NOM_PARA_FONC / INTERPOL_FONC / PROL_DROITE_FONC / PROL_GAUCHE_FONC ... 26
3.25 Opérande INFO ... 26
4 Exemples ... 28
4.1 Calcul d'une enveloppe ... 28
4.2 Calcul de la dérivée de la fonction si ... 28
4.3 Concaténation de deux fonctions ... 28
4.4 Composition de deux fonctions ... 29

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

2 Syntaxe

```
fr          = CALC_FONCTION

     (     ♦     /     DERIVE            =     _F(   ♦     FONCTION    =
     f,                                [fonction]
                                                   ◊     METHODE     =
     'DIFF_CENTREE',   [DEFAUT]
                                                 ),

                 /     INTEGRE     =     _F(   ♦     FONCTION    =     f,
                                [fonction]
                                                   ◊     METHODE     =
     /     'TRAPEZE',  [DEFAUT]

           /     'SIMPSON',
                                                    ◊     COEF
     =     /     0.,                   [DEFAUT]

           /     r,                    [R]
                                                 ),

                 /     INVERSE     =     _F(   ♦     FONCTION    =     f,
                                [fonction]
                                                 ),

                 /     ABS               =     _F(   ♦     FONCTION    =
     f,                                [fonction]
                                                 ),

                 /     ENVELOPPE = _F(   ♦     FONCTION    =     f,
                                [l_fonction]
                                                   ◊     CRITERE     =
     / 'SUP',                [DEFAUT]

           / 'INF',
                                                 ),

                 /     FRACTILE    =     _F(   ♦     FONCTION    =     f,
                                [l_fonction]
                                                   ◊     FRACT       =
     / 1.,                   [DEFAUT]

           / fract                 [R]
                                                 ),
                 /     MOYENNE     =     _F(   ♦     FONCTION    =     f,
                                [l_fonction]
                                                 ),

                 /     COHERENCE   =     _F(   ♦     NAPPE_1           =
     N,                         [NAPPE]
                                ♦     NAPPE_ 2          =     N,
           [NAPPE]
                                ◊     NB_FREQ_LISS      =  / 12 ,
     [DEFAUT]
                                                   / nbl ,           [ I
]
                           ◊     FREQ_COUP   =      FC               [ R]
```


Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

```
                                                    ◊  OPTION      =
                                          / 'TOUT',
      [DEFAUT]
                                                     / '
DUREE_PHASE_FORTE ',
                          SI OPTION  =    ' DUREE_PHASE_FORTE '
                                                    ◊  borne_inf   =
      / 0.05,                 [DEFAUT]
            / binf,                 [R]
                                                    ◊  borne_sup   =
      / 0.95,                 [DEFAUT]
             / BSUP,                [ R ]
                                                  ),

                  /     COMB       =     _F(   ♦    FONCTION    =     f,
                                      [fonction]
                                                  ♦  COEF        =
      r,                              [R]
                                                  ),
                  /     COMB_C           =     _F(   ♦    FONCTION    =
f_c,                                  [fonction_c]
                                                  ♦  /     COEF_R
      = r   ,                             [R]
                                                     /     COEF_C
      = c   ,                             [C]
                                                  ),
                  /     MULT  =     _F(   ♦    FONCTION    =     f,
                                      [fonction]
                                                  ),
                  /     REGR_POLYNOMIALE  =     _F(   ♦    FONCTION    =
f,                [fonction]
                                                                ♦
      DEGRE =     n,                      [I]
                                                  ),
                                  # si COMB ou COMB_C ou REGR_POLYNOMIALE
                                  ◊    LIST_PARA   =     lpara,
                                       [listr8]
                  /     COMPOSE     =     _F(   ♦    FONC_RESU   =
      f_resu      ,                   [fonction]
                                                  ♦  FONC_PARA   =
      f_para      ,                   [fonction]
                                                  ),
                  /     ASSE       =     _F(   ♦    FONCTION    =     l_f,
                                      [l_fonction]
                                                  ◊  SURCHARGE   =
      /     'DROITE',         [DEFAUT]
            /     'GAUCHE',
                                                  ),
```


Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

```
              /     EXTRACTION= _F(    ♦      FONCTION     =      f_c,
              [fonction_c]
                                                ♦      PARTIE
=     /     'REEL',

      /     'IMAG',

      /     'MODULE',

      /     'PHASE',
                                              ),

     / PUISSANCE    =      _F(    ♦      FONCTION     =      f,
                   [fonction]
                                                ◊      EXPOSANT     =
/ n,                    [I]

      / 1,                    [DEFAUT]
                                              ),

              /     FFT               =      _F(    ♦      FONCTION     =
f,                              [fonction]
                                                ◊      METHODE      =
/     'PROL_ZERO',      [DEFAUT]

      /     'TRONCATURE',

      /     'COMPLET',
                                                ◊       SYME        =
/     'OUI',                    [DEFAUT]

      /     'NON',
                                              ),FMIN

              /     CORR_ACCE = _F(     ♦     FONCTION     =      f,
              [fonction]
                                                ♦      METHODE      =
 /    'FILTRAGE',

       /    'POLYNOME',
            si METHODE == 'POLYNOME'
                                                ◊      CORR_DEPL    =
/     'NON',                    [DEFAUT]

      /     'OUI',
            si METHODE == 'FILTRAGE'
                                                ◊      FREQ_FILTRE
=     / fr

            /0.05,                  [DEFAUT]
                                              ),

              /     LISS_ENVELOP      = _F(
                                                ♦       /    NAPPE
=     n,                        [nappe]
                                                /   TABLE
=     t,                        [table]
                              /   FONCTION     =      f,
[fonction]
```


Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

```
                                    ♦ OPTION      =     /     'CONCEPTION',
                                                      / 'VERIFICATION',
                                                        ◊     FREQ_MIN    =
      /     fmin,             [R]
                                                        ◊     FREQ_MAX    =
/     fmax,             [R]
                                                        ◊     ELARG       =
/     elar,             [l_R]
                                                        ◊     LIST_FREQ   =
/     listfreq,         [l_R]
                                    ◊     LIST_AMOR   =     /     listamor,
[l_R]
                                                        ◊     NB_FREQ_LISS
      =     /     nb,           [R]
                                                      = /     10,
 [DEFAUT]
                                                        ◊     ZPA
      =     /     zpa,          [R]
                                                       ),


      /      DSP =      _F(
                                                        ♦     FONCTION          =
      sro ,                    [fonction]
                                                        ♦     AMOR_REDUIT =     lam,
                 [l_R]
                                                         ◊    /     FREQ_PAS    =
      freq_pas,                      [R]
                                                        /     LIST_FREQ   =
      list_freq,         [listr8]
                                                        ♦     FREQ_COUP   =     frc
                         [R]
                                                        ♦     DUREE             =
      tsm                      [R]
                                                        ♦     NORME             =     r,
                               [R]
                                                        ◊     FRACT             =
      /     0.5,               [DEFAUT]

      /     fract              [R]
                                                        ◊     NB_ITER           =
      /     10,                      [DEFAUT]

      /     niter              [R]
                                                 ),

      /     SPEC_OSCI = _F(    ♦     FONCTION          =     f,
            [fonction]
                                                        ◊     METHODE           =
      /'NIGAM',                [DEFAUT]

            /'HARMO'

            /'RICE'
                                                         ◊     TYPE_RESU =      /
      'NAPPE'                  [DEFAUT]
                                                                                 /
      'FONCTION',
```


Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

```
                                                    ◊      AMOR_REDUIT =      lam,
                          [l_R]
                                                    ◊      /      FREQ        =
      lfre,                             [l_R]
                                                           /      LIST_FREQ   =
      lfreq,                            [listr8]
                                                    ◊      NATURE
      =      / 'ACCE',                  [DEFAUT]

             / 'VITE',

             / 'DEPL',
                                                    ♦      NORME              =
      r,                                [R]

si METHODE == 'RICE'
                                                           ◊      NATURE_FONC =
      'DSP',                    [DEFAUT]
                                                           ♦      DUREE              =
      tsm                 [R]
si METHODE == 'NIGAM' ou 'HARMO'
                                                           ◊      NATURE_FONC =
      'ACCE',             [DEFAUT]

                                                    ),

             / PROL_SPEC_OSCI =       _F(    ♦      FONCTION           =      f,
                          [fonction]
                                                    ♦      NORME              =
      r,                                [R]
                                                    ♦      depl_max           =
      dmax,               [DEFAUT]
                                          ),

                    ◊      NOM_PARA                =      para,
                                 [Kn]
                    ◊      NOM_RESU                =      resu,
                                 [K n ]
                    ◊      PROL_DROITE       =      /      'CONSTANT',
                                                           /      'LINEAIRE'
,
                                                           /      'EXCLU',
                    ◊      PROL_GAUCHE       =      /      'CONSTANT',
                                                           /      'LINEAIRE'
,
                                                           /      'EXCLU'
                    ◊      INTERPOL                =      |      'LIN',
                                       [l_Kn]
                                                           |       'LOG',
                    ◊      INTERPOL_FONC           =      |      'LIN',
                                       [l_Kn]
                                                           |      'LOG',
                    ◊      NOM_PARA_FONC           =      parf,
                                 [Kn]
                    ◊      PROL_DROITE_FONC  =      /      'CONSTANT',
                                                           /      'LINEAIRE'
,
                                                           /      'EXCLU',
                    ◊      PROL_GAUCHE_FONC  =      /      'CONSTANT',
```


Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

```
                                                        /     'LINEAIRE'
,
                                                        /     'EXCLU',
                  ◊     INFO  =     /     1,
                                    [DEFAUT]
                                          /     2,
      )
```

Si mot-clé facteur `COMB_C` alors `fr = [FONCTION_C]`,
si mot-clé facteur `SPEC_OSCI` alors `fr = [NAPPE ou FONCTION]` selon les valeurs de `TYPE_RESU` et `AMOR_REDUIT`,
si mot-clé facteur `COHERENCE` alors `fr = [FONCTION]`,
si mot-clé facteur `ENVELOPPE, FRACTILE, PUISSANCE` alors `fr` est du même type que la ou les fonctions en entrée,
pour tous les autres mot-clés facteurs, `fr = [FONCTION]`.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

3 Opérandes

3.1 Mot clé DERIVE

/ DERIVE =

On dérive la fonction $f(t)$.

♦ FONCTION = f

Nom de la fonction que l'on désire dériver.
Ne s'applique pas aux concepts de type nappe.

◊ METHODE =

Nom de la METHODE que l'on désire utiliser : la seule méthode disponible est actuellement DIFF_CENTREE (par défaut).

Remarques :

Voir mot-clé INTEGRE.

3.2 Mot clé INTEGRE

/ INTEGRE =

On intègre la fonction $f(t)$.

◊ COEF = r

Constante d'intégration, par défaut 0.

♦ FONCTION = f

Nom de la fonction que l'on désire intégrer.
Ne s'applique pas aux concepts de type nappe.

◊ METHODE =

Nom de la METHODE que l'on désire utiliser.

Deux méthodes sont disponibles : la méthode des 'TRAPEZE' (par défaut) et la méthode de 'SIMPSON'.

L'intégrale est exacte pour les fonctions linéaires par morceaux telle que fournies en entrée. L'erreur commise par rapport à l'intégrale de la fonction que l'on a discrétisé est en $o(1/n^2)$ pour des fonctions de classes C^2 .

La méthode de Simpson est exacte pour les polynômes de degré inférieur ou égal à 3. L'erreur est en $o(1/n^4)$ pour des fonctions de classes C^4 .

Pour les fonctions quelconques, très chahutées, comme les accélérogrammes, il est conseillé d'utiliser la méthode des trapèzes.
En revanche, quand la fonction $f(t)$ (avant discrétisation) est suffisamment régulière, la méthode de Simpson est bien plus précise.

Remarques :

1) *Pour INTEGRE comme pour DERIVE, le NOM_PARA de la fonction produite est inchangé. En revanche, le NOM_RESU peut être modifié dans les cas suivants : pour la dérivation, DEPL devient VITE, VITE devient ACCE ; pour l'intégration, ACCE devient VITE, VITE devient DEPL. L'utilisateur a toujours la possibilité de le modifier par le mot clé du même nom dans CALC_FONCTION.*

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

2) *Concernant les prolongements, la fonction produite a, par défaut des prolongements* EXCLU *à gauche et à droite quelques soient ceux de la fonction de départ. Ne pas s'attendre donc à ce qu'un prolongement linéaire devienne constant dans la fonction dérivée… Là encore, l'utilisateur est maître de ses prolongements pour la fonction produite par les mots clés* PROL_DROITE *et* PROL_GAUCHE.

3.3 Mot clé INVERSE

/ INVERSE =

On inverse la fonction $f(t)$.

♦ FONCTION = f

Nom de la fonction que l'on désire inverser, il est nécessaire que celle-ci soit bijective (strictement croissante ou strictement décroissante).

Ne s'applique pas aux concepts de type nappe.

Remarque :

1) *Les labels des paramètres ne sont pas inversés ! Le soin est laissé à l'utilisateur d'affecter les valeurs correctes par les mots clés* NOM_PARA *et* NOM_RESU. *Par défaut, le* NOM_PARA *est inchangé et* NOM_RESU *est affecté à* 'TOUTRESU' .
2) *Les modes d'interpolations sont intervertis : par ex. (* 'LIN' , 'LOG' *) devient (* 'LOG' , 'LIN' *).*
3) *Les prolongements* EXCLU *et* LINEAIRE *sont inchangés. En revanche, un prolongement* CONSTANT *est changé en* EXCLU.

3.4 Mot clé ABS

/ ABS =

Fournit la valeur absolue d'une fonction ou d'une nappe.

♦ FONCTION = f

Nom de la fonction dont on désire la valeur absolue.

Remarque :

1) *Les paramètres (prolongements, interpolations,* NOM_PARA *et* NOM_RESU*) de la fonction produite sont les mêmes que ceux de la fonction de départ.*
2) *Sauf pour le prolongement* LINEAIRE *: systématiquement changé en* EXCLU *par précaution. En effet, le prolongement linéaire à droite d'une fonction décroissante conduit pour des abscisses suffisamment grandes à des valeurs négatives : responsabilité est donc laissée à l'utilisateur d'affecter lui-même* PROL_DROITE='LINEAIRE' *(et respectivement à gauche).*

3.5 Mot clé ENVELOPPE

/ ENVELOPPE =

Calcul de l'enveloppe de plusieurs fonctions.

Cette opération est disponible sur des opérandes de nature fonction ou nappe.

3.5.1 Opérande FONCTION

♦ FONCTION = f

Liste des fonctions ou nappes dont on cherche l'enveloppe.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

3.5.2 Opérande CRITERE

◊ CRITERE =

/ 'SUP'

On cherche l'enveloppe supérieure.

/ 'INF'

On cherche l'enveloppe inférieure.

Remarques pour la recherche de l'enveloppe :

- *les fonctions doivent être toutes de même nature (fonction ou nappe),*
- *Cas des fonctions simples : pour les prolongements, interpolations, NOM_PARA et NOM_RESU, ce sont les paramètres de la première des fonctions dans la liste qui sont retenus. Le support d'abscisses de la fonction enveloppe sera la réunion des listes d'abscisses de toutes les fonctions.*
- *Cas des nappes : les paramètres (prolongements, interpolations, NOM_PARA, NOM_RESU, NOM_PARA_FONC) doivent impérativement être identiques entre les nappes fournies. Les supports d'abscisses (valeurs des paramètres et abscisses des fonctions des nappes) sont homogénéisés pour pouvoir calculer l'enveloppe. La nappe produite aura cette discrétisation pour abscisses.*

3.6 Mot clé FRACTILE

/ FRACTILE =

Calcul du fractile de plusieurs fonctions.

Cette opération est disponible sur des opérandes de nature fonction ou nappe.

3.6.1 Opérande FONCTION

♦ FONCTION = f

Liste des fonctions ou nappes dont on cherche à calculer le fractile.

3.6.2 Opérande FRACT

♦ FRACT = fract

Valeur du quantile à calculer. Par défaut fract = 1, le fractile est alors l'enveloppe supérieure.

3.7 Mot clé MOYENNE

/ MOYENNE =

Calcul de la moyenne de plusieurs fonctions.

Cette opération est disponible sur des opérandes de nature fonction ou nappe.

3.8 Mot clé COHERENCE

/ COHERENCE =

Calcul de la fonction de cohérence de signaux temporels.
La fonction de cohérence dépend de la distance entre deux points de mesure et de la fréquence. Elle exprime la corrélation entre les deux signaux, mesurés à une distance d , en fonction de la fréquence. La fonction de cohérence complexe est évaluée en domaine fréquentiel, en passant par les densités spectrales (DSP). L'opérateur renvoie la partie réelle

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

de la cohérence complexe, ce qui correspond à la « *unlagged coherency* » ou la « *plane wave coherency* » dans le cas d'une onde plane.
Les signaux temporels doivent avoir la même discrétisation (même pas de temps dt et même durée T). L'utilisateur doit fournir deux nappes qui contiennent respectivement les signaux aux stations 1 et 2, distancés de d . Les signaux dans chaque nappe sont paramétrés par NUME_ORDRE (1,2,3,..) et ensuite appariés pour calculer les cohérences et en prendre la moyenne.

3.8.1 Opérandes NAPPE_1 et NAPPE_2

♦ NAPPE_1 = nappe1

Nappe contenant N fonctions mesurées à la station 1. Les fonctions de la nappe doivent être numérotées de 1 à N par NOM_PARA = 'NUME_ORDRE'.

♦ NAPPE_2 = nappe2

Nappe contenant N fonctions mesurées à la station 2. Les fonctions de la nappe doivent être numérotées de 1 à N par NOM_PARA = 'NUME_ORDRE'.

3.8.2 Opérande FREQ_COUP

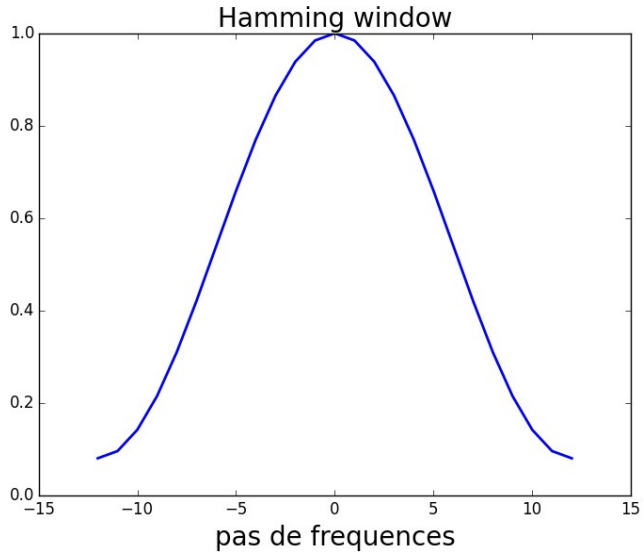
◊ FREQ_COUP = fc

Fréquence de coupure pour le traçage des fonctions de cohérence. La fréquence de coupure doit être inférieure à $\frac{1}{2\times dt}$, où dt est le pas de temps des signaux temporels. Pour l'évaluation des fonctions de cohérence, la fréquence de coupure est toujours déterminée par le pas de temps des signaux temporels. Si FREQ_COUP est renseigné, alors la fonction de cohérence est coupée à fc avant restitution.

3.8.3 Opérande NB_FREQ_LISS

◊ NB_FREQ_LISS = / 12 [defaut]
/ nfl

Nombre de pas de fréquence utilisé pour le lissage des DSP par fenêtre de Hamming. La fenêtre de Hamming obtenue avec NB_FREQ_LISS = 12 est montrée dans la figure ci-dessous.



Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

3.8.4 Opérandes OPTION, BORNE_INF et BORNE_SUP

◊ / OPTION =

Permet de définir si le signal entier ou uniquement la phase forte est utilisé pour le calcul de la fonction de cohérence :

/ 'TOUT' [DEFAUT]

On utilise les signaux temporels tels quels.

/ 'DUREE_PHASE_FORTE'

On évalue la fonction de cohérence pour la durée de phase forte uniquement. Les instants de début et de fin de phase forte sont calculés pour le premier signal uniquement. On utilise ensuite le même intervalle de temps pour les autres signaux. Dans le cas du séisme, cela suppose que tous les enregistrements proviennent du même événement. Pour évaluer la durée de phase forte, on utilise la notion de l'intensité d'Arias. Il faut alors renseigner les bornes inférieure et supérieure (*cf.* [U4.32.05]).

◊ BORNE_INF = /0.05
/binf,
◊ BORNE_SUP = /0.95
/bsup,

Bornes limitant la part d'intensité Arias définissant les instants initial et final de la phase forte (entre $(b_{\inf})\%$ et $(b_{\sup})\%$ de l'intensité d'Arias) du séisme - même signification que pour INFO_FONCTION. Par défaut, on prend 5% et 95%.

3.9 Mot clé COMB et opérande LIST_PARA

/ COMB =

Combinaison linéaire réelle de plusieurs concepts de nature fonction ou nappe.

♦ FONCTION = f

Nom de la fonction à combiner.

♦ COEF = r

Valeur du coefficient.

◊ LIST_PARA = lpara

Liste des valeurs des paramètres pour laquelle la combinaison des fonctions sera discrétisée. Si ce mot clé n'est pas renseigné, une liste par défaut est construite en prenant l'union des listes des valeurs des paramètres de chaque fonction.

Attention :

Ce n'est pas un mot clé du mot clé facteur COMB .

Remarques pour la combinaison :

Voir les remarques pour le mot clé ENVELOPPE

3.10 Mot clé COMB_C et opérande LIST_PARA

/ COMB_C =

Combinaison linéaire complexe de plusieurs concepts de nature fonction_c.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

♦ FONCTION = f_c

Nom de la fonction à combiner. Elle peut être à valeurs complexes ou réelles.

/ COEF_R = r,
/ COEF_C = c,

Valeur du coefficient multiplicateur, soit sous forme réelle r, soit sous forme complexe c.

◊ LIST_PARA = lpara

Liste des valeurs des paramètres pour laquelle la combinaison de fonctions sera discrétisée. Si ce mot clé n'est pas renseigné, une liste par défaut est construite en prenant l'union des listes des valeurs des paramètres de chaque fonction.

Remarques pour la combinaison :

Voir les remarques pour le mot clé ENVELOPPE

3.11 Mot clé MULT et opérande LIST_PARA

/ MULT

Produit de concepts de même nature fonction, fonction_c ou nappe.

♦ FONCTION = f

Nom de la fonction à multiplier avec les autres.

LIST_PARA permet de discrétiser la fonction résultat comme pour COMB.

3.12 Mot clé COMPOSE

Mot clé facteur permettant de calculer la composée de deux fonctions $F(G(t))$.
Ne s'applique pas aux concepts de type nappe.

/ COMPOSE =

♦ FONC_RESU = f_resu

Fonction f_resu(x)

♦ FONC_PARA = f_para

Fonction f_para(t)

On vérifie que le NOM_PARA de f_resu correspond au NOM_RESU de f_para.

3.13 Mot clé ASSE

/ ASSE =

Mot clé facteur permettant de créer une fonction réelle en concaténant deux fonctions réelles tabulées.
Ne s'applique pas aux concepts de type nappe.

3.13.1 Opérande FONCTION

♦ FONCTION = l_f

Fonctions à concaténer. On attend exactement deux fonctions.

3.13.2 Opérande SURCHARGE

◊ SURCHARGE = / 'DROITE',
/ 'GAUCHE',

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

Les points de discrétisation de la fonction créée sont ceux de l'ensemble des deux fonctions, modulo les effets de surcharge.
Si les domaines de définition des fonctions se chevauchent, l'une des fonctions impose ses points sur la zone de recouvrement et pour les prolongements :

SURCHARGE = / 'DROITE' : c'est la fonction qui a le plus grand xmax qui est choisie,
SURCHARGE = / 'GAUCHE' : c'est la fonction qui a le plus petit xmin qui est choisie.

3.13.3 Vérifications

On vérifie que toutes les fonctions ont le même NOM_PARA, ainsi que les mêmes interpolations.

3.14 Mot clé EXTRACTION

/ EXTRACTION =

Mot clé facteur permettant de construire à partir d'une fonction complexe (type fonct_c), une fonction réelle représentant soit la partie réelle, soit la partie imaginaire, soit le module, soit la phase de la fonction complexe.

3.14.1 Opérande FONCTION

♦ FONCTION = f_c

Fonction complexe.

3.14.2 Opérande PARTIE

♦ PARTIE =

/ 'REEL' : extraction de la partie réelle de f_c,
/ 'IMAG' : extraction de la partie imaginaire de f_c,
/ 'MODUL E' : extraction du module de f_c,
/ 'PHASE ' : extraction de la phase (en degré) de f_c.

3.15 Mot clé PUISSANCE

Ce mot clé permet de construire la puissance n^{ième} d'une fonction ou d'un ensemble de fonctions fourni sous forme d'une nappe.

♦ FONCTION = f

Nom de la fonction f concernée (type fonction ou nappe).

♦ EXPOSANT = n

La fonction résultat calculée sera $x \rightarrow f(x)^n$. Par défaut, $n=1$.

3.16 Mot-clé REGR_POLYNOMIALE

Ce mot clé calcule la régression polynomiale d'une fonction par la méthode des moindres carrés (en utilisant la fonction polyfit de numpy).

♦ FONCTION = f

Nom de la fonction f concernée (type fonction).

♦ DEGRE = n

Degré du polynôme recherché.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

On peut utiliser le mot-clé LIST_PARA pour tabuler le polynôme calculé. Sinon, il est tabulé sur la liste des abscisses de la fonction f.

3.17 Mot clé FFT

/ FFT =

On calcule la transformée de Fourier directe ou inverse d'une fonction (dont par l'algorithme FFT).

♦ FONCTION = f

Nom de la fonction sur laquelle s'effectue l'opération.
Si le NOM_PARA de la fonction est INST, alors la FFT directe est calculée.
Si le NOM_PARA de la fonction est FREQ, alors la FFT inverse est calculée.
Ne s'applique pas aux concepts de type nappe.

♦ METHODE =

L'algorithme FFT est plus rapide pour les échantillons dont la longueur est une puissance de 2.
La méthode 'PROL_ZERO' (par défaut) propose de prolonger le signal d'entrée avec des zéros jusqu'à avoir un nombre total d'échantillon qui est la première puissance de 2 dont la valeur est supérieure au nombre d'échantillons initial.
La méthode 'TRONCATURE' ne va considérer que les premiers échantillons dont le nombre total est la plus grande puissance de deux dont la valeur est inférieure au nombre initial d'échantillon.
Par exemple, sur un signal de 601 valeurs, la méthode 'PROL_ZERO' va compléter le signal pour avoir 1024 échantillons, alors que la méthode 'TRONCATURE' ne va considérer que les 512 premiers instants.
Si le signal d'entrée a un nombre d'échantillon qui est une puissance de deux, les deux méthodes sont bien évidemment équivalentes : on prend en compte le signal sans le modifier.
La méthode 'COMPLET' permet de prendre en compte la totalité du signal d'entrée, quelque soit le nombre d'échantillons.
Nota bene : dans le cas d'un échantillon de longueur N, dont le pas de temps serait dt, la fréquence d'échantillonnage de la FFT est $1/(N.dt)$. En revanche, la dernière fréquence pour laquelle la transformée discrète est calculée n'est pas $1/dt$, mais $(N-1)/(N.dt)$.

♦ SYME =

Mot clé qui ne s'applique que pour la transformée de Fourier inverse.
Dans le cas où on fournirait le spectre complet, alors la transformée inverse est calculée directement en utilisant SYME = 'OUI'. Les méthodes 'TRONCATURE' et 'PROL_ZERO' ne sont alors pas actives.
Si le spectre (complexe) fourni en entrée de la FFT inverse ne contient pas la partie repliée (associées aux fréquences négatives du spectre), on peut néanmoins estimer un signal temporel ayant le même contenu spectral sur la partie associée aux fréquences positives. Si on note X_k le k ième échantillon de la transformée de Fourier d'un échantillon de longueur N, alors on a $X_k = X^*_{(N-k)}$, où $()^*$ correspond au complexe conjugué. Cette information peut être exploitée pour reconstruire un signal temporel en ne connaissant que la moitié du spectre. Cette opération est réalisée lorsqu'on choisit SYME = 'NON'. Le signal temporel est alors reconstruit pour obtenir un échantillon temporel de longueur paire. En théorie, pour reconstruire un signal temporel de longueur $2 \times M$, le spectre doit vérifier certaines conditions :

1. Le spectre doit être de longueur $M+1$,
2. Le premier point du spectre doit être réel,
3. Le dernier point du spectre doit être réel.

Si ces conditions ne sont pas vérifiées, alors on construit un spectre approché de longueur impaire vérifiant ces conditions. Si le spectre initial est de longueur paire, le dernier point est alors reconstruit en effectuant un prolongement par interpolation du spectre initial. Cette reconstruction

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

peut introduire un léger biais lorsque le contenu spectral de l'échantillon est très significatif sur les derniers points du spectre.
Les méthodes 'TRONCATURE' et 'PROL_ZERO' sont encore disponibles pour la FFT inverse. Attention, cependant, avec l'utilisation de la méthode 'TRONCATURE'. Si le nombre de point tronqué est significatif, alors les résultats peuvent être très sensiblement différents.

3.18 Mot clé INTERPOL_FFT

/ INTERPOL_FFT =

Interpolation d'un accélérogramme par méthode « zero padding ». Cette méthode consiste à calculer la transformée de Fourier du signal initial, à compléter le signal fréquentiel obtenu par de nouvelles fréquences de valeur nulle puis à calculer la transformée de Fourier inverse de ce signal fréquentiel modifié. Cette fonctionnalité utilise les mêmes algorithmes que le mot-clé FFT. La méthode est nécessairement 'COMPLET' et SYME = 'NON'.
Elle permet de ne pas ajouter de contenu fréquentiel à la fonction interpolée.

♦ FONCTION = f

Nom de la fonction sur laquelle s'effectue l'opération.
NOM_PARA de la fonction est nécessairement INST.

♦ PAS_INST =

Incrément de temps souhaité pour le signal interpolé. Doit être plus petit que l'incrément de temps de la fonction fournie.

◊ PRECISION =
/ prec
/ 0.01 [DEFAUT]

Le pas de temps réellement obtenu est souvent légèrement différent que celui indiqué via PAS_INST. Cet opérande définit l'écart maximum souhaité entre ces deux valeurs, s'il est dépassé, un message d'alarme est émis pour avertir l'utilisateur.

3.19 Mot clé CORR_ACCE

/ CORR_ACCE =

Mot clé facteur permettant de corriger un accélérogramme (signal sismique en accélération) pour que le signal en déplacement ne possède pas de dérive. On retourne en sortie l'accélérogramme corrigé. Le signal peut ensuite être intégré par les méthodes standards (mot-clé INTEGRE de CALC_FONCTION). Il convient de vérifier la qualité du résultat en vérifiant l'absence de dérive du signal en déplacement et en comparant les spectres de réponse avant et après modification.

3.19.1 Opérande FONCTION

♦ FONCTION = f

Accélérogramme réel mesuré.
Ne s'applique pas aux concepts de type nappe.

3.19.2 Opérande METHODE

♦ METHODE =

Nom de la METHODE que l'on souhaite utiliser pour corriger les signaux: correction par POLYNOMES ou par FILTRAGE dans le domaine fréquentiel.

METHODE='POLYNOME'
On supprime la dérive du signal, calculée par lissage linéaire au sens des moindres carrés sur la totalité du signal. La dérive de la vitesse relative correspondante est également supprimée.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

◊ CORR_DEPL =

/ 'NON'

On ne corrige pas la dérive du déplacement relatif, c'est la valeur par défaut.

/ 'OUI'

On supprime aussi la dérive du déplacement relatif. Cette option est à utiliser avec précaution, car on ne connaît pas a priori la valeur du déplacement final après le séisme.

METHODE='FILTRAGE'

On supprime la dérive par filtrage (« passe_haut ») du signal dans le domaine des fréquences. Ce filtre est décrit dans la documentation R4.05.05 (section 2.1).

◊ FREQ_FILTRE =

/ fr

/ 0.05 [DEFAUT]

Il faut choisir la plus petite fréquence qui permette d'obtenir l'effet escompté, à savoir supprimer la dérive, sans (trop) altérer les autres propriétés du signal. Par défaut la fréquence du filtre vaut 0.05Hz. Cette valeur est généralement bien adapté pour les signaux sismiques. Cette valeur peut être diminué si le signal le permet ou augmenté si une dérive persiste.

3.20 Mot clé LISS_ENVELOP

♦ / NAPPE = n

Nom de la nappe ou des nappes d'entrée formée(s) des spectres bruts associés à chaque niveau d'amortissement. La nappe dépend donc des paramètres fréquence et amortissement.

/ TABLE = tab

Nom de la table ou des tables d'entrée formée(s) des spectres bruts associés à chaque niveau d'amortissement. Dans le format attendu, la première colonne contient les valeurs des fréquences, chacune des colonnes suivantes contient la valeur de l'accélération associée à la fréquence pour un niveau d'amortissement donné. Les valeurs des amortissements sont données par LIST_AMOR.

/ FONCTION = n

Nom de la fonction formée du spectre brut associé à un niveau d'amortissement.

♦ OPTION =

/ 'CONCEPTION'

La première étape consiste à réaliser une enveloppe sur les spectres fournis. Dans une seconde étape, on réalise le lissage du spectre obtenu en fonction du nombre de points lissés demandés NB_FREQ_LISS .

/ ' VERIFICATION '

La première étape consiste à réaliser un lissage de chaque spectre fourni selon le nombre de points demandé . Puis, on réalise un élargissement de chaque spectre suivant les coefficients fournis dan s ELARG . Dans une troisième étape, l'enveloppe des spectres élargis est calculée . Enfin la dernière étape réalise le lissage du spectre obtenu en fonction du nombre de points lissés demandé.

◊ NB_FREQ_LISS

Nombre de fréquences souhaitées pour le spectre lissé. Dans le cas de l'option 'CONCEPTION', on ne fournit qu'une valeur. Pour l'option ' VERIFICATION ', on peut fournir deux valeurs qui seront appliquées à la première étape et à la quatrième étape.

◊ FREQ_MIN et FREQ_MAX

Plage de définition en fréquence du spectre lissé.
Les fréquences mentionnées sous FREQ_MIN et FREQ_MAX doivent être choisies parmi les fréquences de discrétisation du spectre brut.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

Par défaut, on considère le spectre complet.

◊ ELARG

L'élargissement porte sur l'ensemble d'un spectre. On doit fournir autant de valeurs que de nombres de nappes ou tables.
Pour chaque fréquence F_i du spectre brut, on définit deux nouvelles valeurs de fréquences telles que :

- $F^- = F_i(1-\tau_g)$ avec $0<\tau_g<1$;
- $F^+ = F_i(1+\tau_d)$ avec $0<\tau_d<1$.

Les paramètres τ_g et τ_d représentent l'amplitude de l'élargissement en fréquence et sont égaux à la valeur fournie.
Les valeurs des fréquences excentrées F^- et F^+ ne correspondent pas aux valeurs F_i de la liste de définition du spectre brut. On définit ainsi F_j et F_k telles que :

- F_j : valeur appartenant à la liste, immédiatement inférieure ou égale à F^- ,
- F_k : valeur appartenant à la liste, immédiatement inférieure ou égale à F^+ .

Pour chaque fréquence F_i , deux points de coordonnées (F_j, γ_i) et (F_k, γ_i) sont définies où γ_i représente l'accélération à la fréquence F_i . Deux nouveaux spectres résultants du décalage du spectre brut sur l'axe des fréquences sont donc construits. Puis on calcule l'enveloppe.

◊ LIST_FREQ

Liste de fréquences que l'on souhaite conserver pour le spectre lissé. Les éléments de cette liste doivent être contenus dans les valeurs de fréquence des spectres pris en entrée.

◊ LIST_AMOR

Liste des amortissements que l'on souhaite associer aux différentes valeurs d'accélérations de la table fournie.

◊ ZPA

Valeur de l'accélération à haute fréquence que l'on souhaite imposer pour le spectre lissé.
Par défaut, il s'agit de la valeur du spectre le moins amorti à la fréquence la plus élevée.

Un exemple d'application est proposé dans le cas test ZZZZ100E.

3.21 Mot clé SPEC_OSCI

/ SPEC_OSCI =

Lorsque METHODE = 'NIGAM' (défaut) ou 'HARMO' (*cf.* §3.21.1) , NATURE_FONC = 'ACCE' (*cf.* §3.21.2), on calcule le spectre d'oscillateur d'un accélérogramme [R4.05.03] donné sous le mot-clé FONCTION (*cf.* §3.21.1).

Le spectre d'oscillateur n'est calculable que sur les fonctions de NOM_RESU = 'ACCE' et de NOM_PARA = 'INST'.

Pour tout i et tout j on considère q_j^i la solution de l'équation différentielle :

$$\ddot{q}_j^i + 2\xi_j\omega_i\dot{q}_j^i + \omega_i^2 q_j^i = f(t)$$

avec $q_j^i(0) = \dot{q}_j^i(0) = f(0)$ et $\omega_i = 2\pi\phi_i$

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

Le concept produit fr est une nappe (fonction à deux variables) constituée des fonctions $\left(fr_i, ..., fr_j, ...\right)$ avec fr_j fonction définie aux points ω_i avec :

$$fr_j(\omega_i)=\underset{t\in D}{Max}\left|q_i^j(t)\right| \text{ et } D=\left\{t / f \text{ définie}\right\}$$

Par défaut pour le calcul du spectre d'oscillateur

- on considère pour les amortissements réduits les valeurs :

  0.02 0.05 0.10

- on considère pour les fréquences, les 150 valeurs suivantes en Hz ,

  la première est à $0.2\,Hz$ et on déduit les suivantes par la règle ;
  de la 2ème à la 57ème : par pas de 0.05 Hz
  58 à 65 : 0.075 Hz
  66 à 79 : 0.10 Hz
  80 à 103 : 0.125 Hz
  104 à 131 : 0.25 Hz
  132 à 137 : 0.5 Hz
  138 à 141 : 1. Hz
  142 à 150 : 1.5 Hz

  La fréquence de coupure par défaut est donc $35,5\,Hz$ (l'utilisateur doit vérifier que cette valeur est cohérente avec le contenu fréquentiel du signal d'entrée, si ce n'est pas le cas, il faut définir une liste de fréquence adaptée).

- le spectre est normé selon la valeur de NORME.

Lorsque METHODE = 'RICE' (*cf.* §3.21.1), NATURE_FONC = 'DSP' (*cf.* §3.21.2), on calcule le spectre d'oscillateur équivalent à une densité spectrale de puissance (DSP) [R4.05.03] sous le mot-clé FONCTION (*cf.* §3.21.1). La fonction doit être de NOM_PARA = 'FREQ'.

3.21.1 Opérande FONCTION

♦ FONCTION = f

Nom de la fonction sur laquelle s'effectue l'opération.
Ne s'applique pas aux concepts de type nappe.

3.21.2 Opérandes NATURE / NATURE_FONC

◊ NATURE =

Nature de la grandeur de la nappe créée par la commande CALC_FONCTION.

ACCE	Spectre de pseudo-accélération	$\ddot{u}(t)=\omega_i^2 u(t)$
VITE	Spectre de pseudo-accélération	$\dot{u}(t)=\omega_i u(t)$
DEPL	Spectre de pseudo-accélération	$u(t)$

◊ NATURE_FONC = / 'ACCE'

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

/ 'DSP'

Nature de la fonction qui sert à construire le spectre. Le choix est imposé selon la méthode sélectionnée : 'ACCE' (signal temporel) ou 'DSP' (densité spectrale de puissance en accélération). Ce mot clé permet de surcharger le NOM_RESU de la fonction spécifiée sous le mot clé FONCTION lorsque celle-ci est créée par RECU_FONCTION [U4.32.03].

3.21.3 Opérande METHODE

◊ METHODE =

Nom de la METHODE que l'on désire utiliser : la méthode par défaut, 'NIGAM', est détaillée dans le document [R5.05.01]. Si on choisit la méthode 'HARMO', alors le spectre de réponse est obtenu par des calculs harmoniques successifs (pour différentes fréquences propres d'oscillateur) en passant par une FFT/IFFT du signal en entrée. Si on choisit 'RICE' alors on détermine le SRO équivalent pour la donnée d'une DSP.

3.21.4 Opérande TYPE_RESU

◊ TYPE_RESU = / 'NAPPE' [DEFAUT]
/ 'FONCTION'

Par défaut, l'utilisation de SPEC_OSCI produit une nappe. Si AMOR_REDUIT ne contient qu'un amortissement, on crée une nappe ne contenant qu'une fonction. Dans ce cas, on peut choisir TYPE_RESU='FONCTION' pour retourner directement cette fonction.

3.21.5 Opérande AMOR_REDUIT

◊ AMOR_REDUIT = lam

lam = $\left(\xi_1, ..., \xi_i, ...\right)$

Liste des amortissements réduits : exemple 0.01, 0.05, …

3.21.6 Opérandes FREQ / LIST_FREQ

~ ◊ FREQ = lfre

lfre = $\left(\phi_1, ..., \phi_i, ...\right)$. Liste des fréquences.

~ ◊ LIST_FREQ = lfreq

Liste des fréquences fournies sous un concept listr8.

Les fréquences doivent être strictement positives.

3.21.7 Opérande NORME

♦ NORME = r

Le spectre d'oscillateur sera normé à la valeur r (valeur de la pseudo-accélération), cette valeur est rappelée dans le fichier de message.

3.21.8 Opérande DUREE

♦ DUREE = tsm

Ce mot-clé est à renseigner uniquement si METHODE='RICE' et donc NATURE_FONC='DSP'. Il s'agit de la durée de la phase forte du séisme. Le signal sismique est considéré comme stationnaire sur cette durée. Cette valeur est nécessaire pour évaluer le facteur de pic qui intervient dans la formule de Vanmarcke pour trouver le SRO équivalent à la donnée de la DSP.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

3.22 Mot clé PROL_SPEC_OSCI

/ SPEC_OSCI = _F(

Ce mot-clé permet de prolonger un spectre en dessous de la fréquence de définition (par exemple fmin = 0.25Hz) en donnant comme critère une valeur de déplacement maximal dmax.

On calcule la pente à partir des deux premières valeurs du spectre (la plus petite fréquence fmin est supérieure à zéro) et on prolonge le spectre en déplacement en retenant cette pente jusqu'à la valeur dmax.

3.22.1 Opérande FONCTION

♦ FONCTION = f

Nom de la fonction sur laquelle s'effectue l'opération. Ne s'applique pas aux concepts de type nappe.

3.22.2 Opérande NORME

♦ NORME = r

Le spectre d'oscillateur sera normé à la valeur r (valeur de la pseudo-accélération), cette valeur est rappelée dans le fichier de message.

3.22.3 Opérande DEPL_MAX

♦ DEPL_MAX = dmax

Valeur de déplacement utilisée pour prolonger le spectre.
Si la pente du spectre est inférieure à zéro, il faut que dmax soit plus grand que la première valeur de déplacement du spectre donné.

3.23 Mot clé DSP

/ DSP =

Calcule la densité spectrale de puissance (DSP) équivalente à la donnée d'un spectre de réponse d'oscillateur (SRO) avec la formule de Vanmarcke, fonction de nature fonction [R4.05.03].

Le spectre d'oscillateur n'est calculable que sur les fonctions de NOM_PARA = 'FREQ'.

Pour le calcul de la DSP à partir d'un spectre d'oscillateur, on considère que

- le SRO exprime le maximum médian (fractile 0.5), sinon il faut renseigner une autre valeur via le mot clé FRACT.
- la DSP vaut zéro pour des fréquences inférieures ou égales à $1/2\pi\,Hz$.
- le spectre est normé selon la valeur de NORME.

L'utilisateur doit vérifier que la discrétisation fréquentielle (liste des fréquences) est suffisante par rapport au contenu fréquentiel des signaux à modéliser. Il convient également de vérifier l'équivalence entre le DSP et le SRO donnée en faisant des tirages ou en déterminant des valeurs de réponse maximale d'un oscillateur avec POST_DYNA_ALEA. L'opérateur calcule la densité spectrale de puissance (DSP) équivalente à la donnée d'un spectre de réponse d'oscillateur (SRO) avec la formule de Vanmarcke. On n'effectue pas d'itération pour optimiser le fit.
Un exemple d'application est proposé dans le cas test ZZZZ100D.

3.23.1 Opérande FONCTION

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

♦ FONCTION = sro

Nom de la fonction définissant le SRO.

3.23.2 Opérandes AMOR_REDUIT, LIST_FREQ, FREQ_PAS

Les mot-clés AMOR_REDUIT et LIST_FREQ sont identiques à ceux décrits pour SPEC_OSCI (cf. 3.21).
Le mot-clé FREQ_PAS désigne le pas de fréquence si LIST_FREQ n'est pas renseigné.

3.23.3 Opérandes FREQ_COUP

◊ FREQ_COUP = frc

La fréquence de coupure : on détermine la DSP jusque cette fréquence. Le SRO est prolongé (valeur constante correspondante à la ZPA) jusqu'à cette valeur si nécessaire.

3.23.4 Opérande DUREE

♦ DUREE = tsm

Durée de la phase forte du séisme. Le signal sismique est considéré comme stationnaire sur cette durée. Cette valeur est nécessaire pour évaluer le facteur de pic qui intervient dans la formule de Vanmarcke.

3.23.5 Opérande NORME

♦ NORME = r

On considère des spectres d'oscillateur normés à la valeur r (valeur de la pseudo-accélération). En général, les SRO sont donnés en g, il faut donc renseigner NORME = 9.81.

3.23.6 Opérande FRACT

◊ FRACT = 0.5 [DEFAUT]
fract

La DSP est ajustée sur un fractile du spectre de réponse. La valeur du fractile a un impact sur la détermination du facteur de pic, utilisé pour dans la formule de Vanmarcke.En général on considère que le spectre cible correspond à la médiane et donc fract = 0.5.

3.23.7 Opérande NB_ITER

◊ NB_ITER = / 10 [DEFAUT]
/ niter

Nombre d'itérations pour améliorer l'ajustement de la DSP. Il peut être nécessaire de réaliser moins d'itérations afin d'éviter l'*overfitting* de la DSP (deux itérations par exemple).

3.24 Attributs du concept fonction en sortie

3.24.1 Valeurs par défaut

Par défaut les attributs du concept fonction en sortie de la commande CALC_FONCTION sont pour les différentes options (cf. commandes DEFI_FONCTION [U4.31.02] et DEFI_NAPPE [U4.31.03]).

- Option DERIVE :

Interpolation : donnée par la fonction en entrée

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

Prolongement gauche : EXCLU

Prolongement droit : EXCLU

NOM_PARA = 'INST' (exemple) donné par la fonction en entrée

NOM_RESU = 'VITE' (exemple) donné par la fonction en entrée

- Option INTEGRE :

Même règles que pour DERIVE

- Options COMB / COMB_C :

Les attributs de la première fonction combinée.

- Option SPEC_OSCI : le résultat est une nappe sauf si TYPE_RESU='FONCTION' et qu'il n'y a qu'une valeur d'amortissement réduit.

Les attributs de la nappe :

NOM_PARA = 'AMOR'

NOM_RESU = 'DEPL' ou 'VITE' ou 'ACCE'

Interpolation : 'LOG'

Prolongement gauche : 'EXCLU'

Prolongement droit : 'EXCLU'

Les attributs de chaque fonction :

NOM_PARA = 'FREQ'

Interpolation : 'LOG'

Prolongement gauche : 'EXCLU'

Prolongement droit : 'CONSTANT'

- Option ENVELOPPE :

Les attributs de la première fonction donnée.

- Option FFT :

NOM_PARA = FREQ si NOM_PARA de la fonction est INST
Sinon c'est l'inverse

- Option COMPOSE :

NOM_PARA : celui de la fonction FONC_PARA
NOM_RESU : celui de la fonction FONC_RESU
INTERPOL : celui de la fonction FONC_RESU
Prolongement : celui de la fonction FONC_RESU

- Option EXTRACTION :

Attributs identiques à ceux de la fonction donnée en entrée

- Option ASSE :

NOM_PARA : celui des fonctions
NOM_RESU : celui des fonctions
INTERPOL : linéaire
Prolongement : 'EXCLU'

3.24.2 Surcharge des attributs

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

3.24.2.1 Opérande NOM_PARA

◊ NOM_PARA = para

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

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

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

3.24.2.2 Opérande NOM_RESU

◊ NOM_RESU = resu

Il permet de documenter, la fonction créée en donnant un nom (8 caractères) à la fonction. Sauf exception (cf. [§3.1], [§3.2], [§3.5]), ce nom n'est pas testé.

3.24.2.3 Opérande INTERPOL

◊ INTERPOL

Quand le concept produit est une nappe, INTERPOL définit le type d'interpolation entre deux valeurs consécutives du paramètre de la nappe et entre deux fonctions (une fois celles-ci évaluées). Même sens que dans DEFI_NAPPE.
Quand le concept produit est une fonction (réelle ou complexe), il définit le type d'interpolation pour les abscisses et les ordonnées de la fonction. On attend jusqu'à deux valeurs. S'il une seule valeur est fournie, elle est utilisée pour les abscisses et les ordonnées. Même sens que dans DEFI_FONCTION.

3.24.2.4 Opérandes PROL_DROITE / PROL_GAUCHE

◊ PROL_DROITE et PROL_GAUCHE

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

- 'CONSTANT' pour un prolongement avec la dernière (ou la 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.

3.24.2.5 Opérandes NOM_PARA_FONC / INTERPOL_FONC / PROL_DROITE_FONC / PROL_GAUCHE_FONC

Ces mots clés permettent de modifier les attributs de la nappe et s'appliquent aux paramètres des fonctions de celle-ci. Ils ont la même signification que les mots clés sans le suffixe FONC.

- NOM_PARA_FONC est le nom du paramètre des fonctions (comme dans DEFI_NAPPE).
- INTERPOL_FONC est le type d'interpolation pour les abscisses et ordonnées des fonctions de la nappe (identique au mot-clé INTERPOL du mot-clé facteur DEFI_FONCTION de DEFI_NAPPE).
- PROL_GAUCHE_FONC/PROL_DROITE_FONC définissent les prolongements des fonctions (identiques aux mots-clés PROL_GAUCHE/PROL_DROITE du mot-clé facteur DEFI_FONCTION de DEFI_NAPPE).

3.25 Opérande INFO

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

◊ INFO

Si INFO=2, on imprime la fonction (IMPR_FONCTION format TABLEAU) dans le fichier MESSAGE.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

4 Exemples

4.1 Calcul d'une enveloppe

Le fichier de commandes qui suit :

```
DEPI=2. * pi
PAS0=DEPI / 200.
LI1=DEFI_LIST_REEL(    DEBUT=0.,
                                    INTERVALLE=_F(    JUSQU_A  =  DEPI,  PAS  =
PAS0) )

COa = FORMULE(NOM_PARA='INST',VALE='cos(INST)')
SIa = FORMULE(NOM_PARA='INST',VALE='sin(INST)')

CO  = CALC_FONC_INTERP(     FONCTION=COa,    LIST_PARA=LI1,
                            NOM_PARA='INST',
                            NOM_RESU='DEPL',
                            PROL_GAUCHE='EXCLU', PROL_DROITE='LINEAIRE',
                            INTERPOL='LIN',
                            TITRE=' FONCTION COSINUS'   )

SI  = CALC_FONC_INTERP(     FONCTION=SIa,    LIST_PARA=LI1,
                            NOM_PARA='INST',
                            NOM_RESU='DEPLACEMENT',
                            PROL_GAUCHE='EXCLU',
                            PROL_DROITE='CONSTANT',
                            INTERPOL='LIN',
                            TITRE=' FONCTION SINUS '    )

ENV1=CALC_FONCTION( ENVELOPPE=_F(     FONCTION = ( SI,  CO, ),
                                                    CRITERE = 'SUP') )
```

4.2 Calcul de la dérivée de la fonction `si`

Les commandes qui suivent

```
           der1 = CALC_FONCTION( DERIVE=_F( FONCTION= si ),)

           inst1 = 20. * pas

           TEST_FONCTION(    VALEUR=
                                              _F(   FONCTION = der1,
NOM_PARA = 'inst',
                                                    VALE_PARA= inst1,
VALE_REFE= COa(inst1),)
                                    )
```

produisent sur le fichier `'RESULTAT'` :

```
     ---- FONCTION : DER1
     OK  INST         RELA    -0.016 %   VALE: 8.0888392298046D-01
      6.28319E-01 TOLE      0.100 %      REFE: 8.0901699437495D-01
```

4.3 Concaténation de deux fonctions

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

```
DFC1=DEFI_FONCTION(    NOM_PARA='X',   NOM_RESU='Y',
                                   VALE=(      0., 10.,
                                               4., 14.,
                                               6., 16.,),
                                   PROL_DROITE='LINEAIRE',
                                   PROL_GAUCHE='LINEAIRE'
                       )

#

DFC2=DEFI_FONCTION(    NOM_PARA='X',   NOM_RESU='Y',
                                   VALE=(      5., 25.,
                                               7., 27.,
                                               8., 28.,),
                                   PROL_DROITE='LINEAIRE',
                                   PROL_GAUCHE='LINEAIRE'
                       )

#

DFC3=CALC_FONCTION(    ASSE=_F(
                                   FONCTION = ( DFC2,  DFC1, ),
                                   SURCHARGE = 'DROITE')
                       )

DFC4=CALC_FONCTION(    ASSE=_F(
                                   FONCTION = ( DFC1,  DFC2, ),
                                   SURCHARGE = 'GAUCHE')
                       )
```

Les valeurs de la fonction `dfc3` sont :

```
            x     =     0.          4.          5.          7.
 8.
            y     =     10.        14.         25.         27.
28.
```

Les valeurs de la fonction `dfc4` sont :

```
            x     =     0.          4.          6.          7.
 8.
            y     =     10.        14.         16.         27.
28.
```

4.4 Composition de deux fonctions

```
fonc1 = DEFI_FONCTION   (     NOM_PARA    =     'X',
                                    NOM_RESU    =     'F',
                                    VALE        =     (     0.,
      0.,

    2.,          5.,

    3.,         10.,

    5.,         15.,

    7.,         13.,
```


Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

```
      8.,         10.,

      10.,   9.,

      12.,   8.,

      13.,   5.,

      15.,   1.,

      20.,   0.         )     )
fonc2 = DEFI_FONCTION   (     NOM_PARA    =     'INST',
                                                NOM_RESU    =     'X',
                                                VALE        =     (     0.,
            0.,

      0.1,   2.,

      0.2   ,      4.,

      0.3   ,      6.,

      0.4   ,      8.,

      0.5   ,     10.,

      0.6   ,     12.,

      0.7   ,     14.,

      0.8   ,     16.,

      0.9   ,     18.,

      1.0   ,     20.         )     )
comp1 = CALC_FONCTION   (     COMPOSE =   _F(   FONC_RESU = fonc1,

      FONC_PARA = fonc2 )
                                            )
```

Les valeurs de la fonction `comp1` sont :

```
inst =     0.    0.1          0.2   0.3   0.4  0.5         0.6         0.7
           0.8         0.9          1.0
F          =     0.    5.           12.5  14.         10.         9.
      8.           3.           0.8         0.4         0.
```


Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)