

Opérateur CALC_FONCTION

1 But

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

Les opérations suivantes sont actuellement disponibles :

- la dérivation d'une fonction,
- l'intégration d'une fonction,
- l'inverse d'une fonction,
- la valeur absolue d'une fonction,
- la recherche de l'enveloppe de plusieurs fonctions,
- le calcul du fractile de nappes ou de fonctions,
- la combinaison linéaire réelle ou complexe de plusieurs fonctions,
- la composition de deux fonctions,
- la concaténation (mise bout à bout avec gestion des chevauchements) de plusieurs fonctions,
- l'extraction d'une fonction réelle à partir d'une fonction complexe,
- le calcul de la puissance $n^{\text{ième}}$ d'une fonction,
- le calcul de FFT directe ou inverse d'une fonction,
- la correction d'un accélérogramme mesuré en vue du calcul d'une réponse sismique,
- le lissage enveloppe d'un ou plusieurs spectres bruts d'oscillateur,
- le calcul du spectre d'oscillateur d'un accélérogramme (fonction de la fréquence et de l'amortissement) sous forme d'une nappe,

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.

Table des matières

1 But.....	1
2 Syntaxe.....	4
3 Opérandes.....	6
3.1 Mot clé DERIVE.....	6
3.2 Mot clé INTEGRE	6
3.3 Mot clé INVERSE.....	7
3.4 Mot clé ABS.....	7
3.5 Mot clé ENVELOPPE.....	7
3.5.1 Opérande FONCTION.....	7
3.5.2 Opérande CRITERE.....	8
3.6 Mot clé FRACTILE.....	8
3.6.1 Opérande FONCTION.....	8
3.6.2 Opérande FRACT.....	8
3.7 Mot clé COMB et opérande LIST_PARA.....	8
3.8 Mot clé COMB_C et opérande LIST_PARA.....	9
3.9 Mot clé COMPOSE.....	9
3.10 Mot clé ASSE.....	9
3.10.1 Opérande FONCTION.....	9
3.10.2 Opérande SURCHARGE.....	9
3.10.3 Vérifications.....	10
3.11 Mot clé EXTRACTION.....	10
3.11.1 Opérande FONCTION.....	10
3.11.2 Opérande PARTIE.....	10
3.12 Mot clé PUISSANCE.....	10
3.13 Mot clé FFT.....	10
3.14 Mot clé CORR_ACCE.....	11
3.14.1 Opérande FONCTION.....	11
3.14.2 Opérande CORR_DEPL.....	11
3.15 Mot clé LISS_ENVELOP.....	11
3.16 Mot clé SPEC_OSCI.....	12
3.16.1 Opérande FONCTION.....	13
3.16.2 Opérande METHODE.....	13
3.16.3 Opérande AMOR_REDUIT.....	13
3.16.4 Opérandes FREQ / LIST_FREQ.....	13
3.16.5 Opérandes NATURE / NATURE_FONC.....	13
3.16.6 Opérande NORME.....	14
3.17 Attributs du concept fonction en sortie.....	14
3.17.1 Valeurs par défaut.....	14
3.17.2 Surcharge des attributs.....	15
3.17.2.1 Opérande NOM_PARA.....	15

3.17.2.2 Opérande NOM_RESU.....	15
3.17.2.3 Opérande INTERPOL.....	15
3.17.2.4 Opérandes PROL_DROITE / PROL_GAUCHE.....	15
3.17.2.5 Opérandes NOM_PARA_FONC / INTERPOL_FONC / PROL_DROITE_FONC / PROL_GAUCHE_FONC.....	15
3.18 Opérande INFO.....	16
4 Exemples.....	17
4.1 Calcul d'une enveloppe.....	17
4.2 Calcul de la dérivée de la fonction si.....	17
4.3 Concaténation de deux fonctions.....	18
4.4 Composition de deux fonctions.....	19

2 Syntaxe

```

fr      = CALC_FONCTION

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

/ INTEGRE = _F( ♦ FONCTION = f, [fonction]
                ♦ METHODE = / 'TRAPEZE', [DEFAULT]
                        / 'SIMPSON',
                ♦ COEF = / 0., [DEFAULT]
                        / r, [R]
                ),

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

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

/ ENVELOPPE = _F( ♦ FONCTION = f, [l_fonction]
                  ♦ CRITERE = / 'SUP', [DEFAULT]
                        / 'INF',
                  ),

/ FRACTILE = _F( ♦ FONCTION = f, [l_fonction]
                  ♦ FRACT = / 1., [DEFAULT]
                        / fract [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]
                ),
    # si COMB ou COMB_C
    ♦ 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', [DEFAULT]
                        / 'GAUCHE',
              ),

/ EXTRACTION= _F( ♦ FONCTION = f_c, [fonction_c]
                   ♦ PARTIE = / 'REEL',
                           / 'IMAG',
                           / 'MODULE',
                           / 'PHASE',
                   ),

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

```

```

/ 1, [DEFAULT]
),
/ FFT = _F(♦ FONCTION = f, [fonction]
           ◊ METHODE = / 'PROL_ZERO', [DEFAULT]
                       / 'TRONCATURE',
                       / 'COMPLET',
           ◊ SYME = / 'OUI', [DEFAULT]
                  / 'NON',
           ),
/ CORR_ACCE = _F(♦ FONCTION = f, [fonction]
                  ◊ CORR_DEPL = / 'NON', [DEFAULT]
                           / 'OUI',
                  ),
/ LISS_ENVELOP = _F(
  ♦ NAPPE = n, [nappe]
  ◊ FREQ_MIN = / fmin, [R]
                / 0.2 [DEFAULT]
  ◊ FREQ_MAX = / fmax, [R]
                / 35.5 [DEFAULT]
  ◊ ELARG = / elar, [R]
            / 0.1, [DEFAULT]
  ◊ TOLE_LISS = / toleliss, [R]
                / 0.25, [DEFAULT]
  ),
/ SPEC_OSCI = _F(♦ FONCTION = f, [fonction]
                  ◊ METHODE = 'NIGAM', [DEFAULT]
                  ◊ AMOR_REDUIT = lam, [1_R]
                  ◊ / FREQ = lfre, [1_R]
                    / LIST_FREQ = lfreq, [listr 8]
                  ◊ NATURE = / 'ACCE', [DEFAULT]
                           / 'VITE',
                           / 'DEPL',
                  ◊ NATURE_FONC = 'ACCE', [DEFAULT]
                  ♦ NORME = r, [R]
                  ),
◊ NOM_PARA = para, [Kn]
◊ NOM_RESU = resu, [K n ]
◊ PROL_DROITE = / 'CONSTANT',
                / 'LINEAIRE',
                / 'EXCLU',
◊ PROL_GAUCHE = / 'CONSTANT',
                / 'LINEAIRE',
                / 'EXCLU'
◊ INTERPOL = | 'LIN', [1_Kn]
              | 'LOG',
              | 'NON',
◊ INTERPOL_FONC = | 'LIN', [1_Kn]
                  | 'LOG',
                  | 'NON',
◊ NOM_PARA_FONC = parf, [Kn]
◊ PROL_DROITE_FONC = / 'CONSTANT',
                    / 'LINEAIRE',
                    / 'EXCLU',
◊ PROL_GAUCHE_FONC = / 'CONSTANT',
                    / 'LINEAIRE',
                    / 'EXCLU',

```

```
        ◇ INFO = / 1,  
                / 2,  
    )
```

Si mot-clé facteur COMB_C alors fr = [FONCTION_C],
si mot-clé facteur SPEC_OSCI alors fr = [NAPPE],
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-clé facteur, fr = [FONCTION].

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 $f(t)$.

◆ 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'`.

La méthode de `'SIMPSON'` est à employer avec précaution car elle peut entraîner des oscillations. Il vaut mieux discrétiser finement $f(t)$ et intégrer avec la méthode des `'TRAPEZE'`. Notamment il est déconseillé d'utiliser la méthode `'SIMPSON'` pour l'interprétation d'un accélérogramme.

◇ COEF = r

Constante d'intégration, par défaut 0.

Remarques :

- 1) Pour *INTEGRE* comme pour *DERIVE*, le *NOM_PARA* de la fonction produite est inchangé : il ne faut par exemple pas s'attendre à que le *NOM_RESU*='ACCE' produise *NOM_RESU*='VITE' dans la fonction intégrée. L'utilisateur a la possibilité de le modifier par le mot clé du même nom dans *CALC_FONCTION*.
- 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.

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 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*

3.8 Mot clé COMB_C et opérande LIST_PARA

/ COMB_C =

Combinaison linéaire complexe de plusieurs concepts de nature *fonction_c*.

♦ 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

3.9 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 $f_{\text{resu}}(x)$

♦ FONC_PARA = f_para
Fonction $f_{\text{para}}(t)$

On vérifie que le NOM_PARA de f_{resu} correspond au NOM_RESU de f_{para} .

3.10 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.

3.10.1 Opérande FONCTION

♦ FONCTION = l_f

Fonctions à concaténer. On attend exactement deux fonctions.

3.10.2 Opérande SURCHARGE

◇ SURCHARGE = / 'DROITE',
/ 'GAUCHE',

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 x_{max} qui est choisie,
SURCHARGE = / 'GAUCHE' : c'est la fonction qui a le plus petit x_{min} qui est choisie.

3.10.3 Vérifications

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

3.11 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.

3.11.1 Opérande FONCTION

◆ `FONCTION = f_c`

Fonction complexe.

3.11.2 Opérande PARTIE

◆ `PARTIE =`

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

3.12 Mot clé PUISSANCE

Ce mot clé permet de construire la puissance $n^{\text{ième}}$ 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, $n = 1$.

3.13 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.

NB : dans le cas d'un échantillon de longueur N , dont le pas de temps serait dt , la fréquence d'échantillonnage de la FFT est $1/(N \cdot dt)$. En revanche, la dernière fréquence pour laquelle la transformée discrète est calculée n'est pas $1/dt$, mais $(N-1)/(N \cdot dt)$.

♦ 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 $k^{\text{ième}}$ 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 $2 \cdot M$, 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 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.

3.14 Mot clé CORR_ACCE

/ CORR_ACCE =

Mot clé facteur permettant de corriger un accélérogramme mesuré en vue du calcul de la réponse sismique d'un système.

On supprime la dérive du signal, calculée par lissage linéaire au sens des moindres carrés sur la totalité du signal, afin de rendre l'accélérogramme plus réaliste. La dérive de la vitesse relative correspondante est également supprimée.

On retourne en sortie l'accélérogramme corrigé.

3.14.1 Opérande FONCTION

♦ FONCTION = f

Accélérogramme réel mesuré.

Ne s'applique pas aux concepts de type *nappe*.

3.14.2 Opérande CORR_DEPL

◇ 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.

3.15 Mot clé LISS_ENVELOP

Les données d'origine sont constituées d'une nappe de spectres SRO brut définis sur un grand nombre de points pour un niveau de plancher donné.

La première étape consiste, pour chaque spectre, à l'élargissement en fréquence (décalage à gauche et à droite) suivi d'une diminution du nombre de point de définition. Ces opérations effectuées, on s'assure du caractère enveloppe du spectre lissé par rapport au spectre initial. A ce stade, chaque spectre possède sa propre base de fréquence.

La seconde étape consiste à homogénéiser la base de fréquence de l'ensemble des spectres de la nappe en s'assurant du non recouvrement des spectres entre eux.

♦ NAPPE = n

Nom de la nappe d'entrée formée des spectres bruts associés à chaque niveau d'amortissement.

◇ 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.

Par défaut, on considère le spectre complet.

◇ ELARG

L'élargissement porte sur l'ensemble du spectre,

Il est donné en pourcentage et vaut 0.1 (10%) par défaut.

Pour chaque fréquence F_i du spectre brut, on définit deux nouvelles valeurs de fréquences telles que :

$$\bullet \quad F^- = F_i(1 - \tau_g) \quad \text{avec} \quad 0 < \tau_g < 1 \quad ;$$

$$\bullet \quad F^+ = F_i(1 + \tau_d) \quad \text{avec} \quad 0 < \tau_d < 1 \quad .$$

Les paramètres τ_g et τ_d représentent l'amplitude de l'élargissement en fréquence.

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.

◇ TOLE_LISS

Critère en pourcentage portant sur l'élimination des points lors de lissage. Cette tolérance est fixée à 0.25 fois la valeur par défaut.

Le lissage est réalisé sur l'enveloppe des spectres brut, décalés à droite et à gauche.

Un exemple d'application est proposé dans le cas test ZZZZ100e.

3.16 Mot clé SPEC_OSCI

/ SPEC_OSCI =

Calcule le spectre d'oscillateur d'un accélérogramme, fonction de nature fonction [R4.05.03].

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 :

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

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

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

$$fr_j(\omega_i) = \max_{t \in D} |q_j^i(t)| \quad \text{et} \quad D = \{t / f \text{ définie}\}$$

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 65 0.075 Hz
 66 79 0.10 Hz
 80 103 0.125 Hz
 104 131 0.25 Hz
 132 137 0.5 Hz
 138 141 1. Hz
 142 150 1.5 Hz
- le spectre est normé selon la valeur de NORME.

3.16.1 Opérande FONCTION

- ♦ $FONCTION = f$
Nom de la fonction sur laquelle s'effectue l'opération.
Ne s'applique pas aux concepts de type `nappe`.

3.16.2 Opérande METHODE

- ◇ $METHODE =$
Nom de la `METHODE` que l'on désire utiliser : la seule méthode utilisable actuellement est 'NIGAM' (par défaut) qui est détaillée dans le document [R5.05.01].

3.16.3 Opérande AMOR_REDUIT

- ◇ $AMOR_REDUIT = lam$
 $lam = (\xi_1, \dots, \xi_i, \dots)$
Liste des amortissements réduits : exemple 0.01, 0.05, ...

3.16.4 Opérandes FREQ / LIST_FREQ

- ~ ◇ $FREQ = lfre$
 $lfre = (\varphi_1, \dots, \varphi_i, \dots)$. Liste des fréquences.
- ~ ◇ $LIST_FREQ = lfreq$
Liste des fréquences fournies sous un concept `listr8`.

Les fréquences doivent être strictement positives.

3.16.5 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'

Nature de la fonction qui sert à construire le spectre. Pour l'instant seule la valeur 'ACCE' est disponible. 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].

3.16.6 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.

3.17 Attributs du concept fonction en sortie

3.17.1 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
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
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'

3.17.2 Surcharge des attributs

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

3.17.2.1 Opérande NOM_PARA

◇ NOM_PARA = para

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'		

3.17.2.2 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.

3.17.2.3 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 'NON', '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.

3.17.2.4 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.

3.17.2.5 Opérandes NOM_PARA_FONC / INTERPOL_FONC / PROL_DROITE_FONC / PROL_GAUCHE_FONC

Ces mots clés permettent de modifier les attributs des fonctions qui interviennent dans la définition des nappes. Ils ont donc la même signification que les mots clé sans le suffixe FONC.

3.18 Opérande INFO

◇ INFO

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

4 Exemples

4.1 Calcul d'une enveloppe

Le fichier de commandes qui suit :

```
DEPI=2. * pi
PAS0=DEPI / 200.
LI11=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=LI11,
                            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=LI11,
                            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

```
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.,
                                     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.