Operator CALC_FONCTION

1 Drank

To carry out mathematical operations on data structures of type function.

The following operations are currently available :

- the derivative of a function,
- the integration of a function,
- the reverse of a function,
- the absolute value of a function,
- the search of the envelope of several functions,
- the computation of the fractile of three-dimensions functions or functions,
- the real or complex linear combination several functions,
- the composition of two functions,
- the product of functions,
- the concatenation (put end to end with management of the overlappings) several functions,
- the extraction of a real function from a function complex,
- the computation of the power n ième of a function,
- the polynomial regression of a function,
- the computation of FFT direct or opposite of a function,
- the correction of one accelerogram measured for computation of a seismic response,
- the lissage wraps one or more rough spectrums of oscillator,
- the computation of the oscillator spectrum of an accelerogram (function of the frequency and damping) in the form of a three-dimensions function,
- the computation of a function of spectral concentration equivalent to the data of one oscillator spectrum using the formula of Vanmarcke.

Product a data structure `function`, `fonction_c` or `three-dimensions function`, according to the key word factor used.
In output of the command, the function is reordered by increasing X-coordinates.

Contents

2 Syntax

```
Fr      = CALC_FONCTION

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

          /INTEGRE     =_F  ( ♦fonction=f         ,
[function]
                              ◊METHODE=/             "TRAPEZE",
[DEFAULT]
                                                   /   "SIMPSON",
                               ◊   COEFF         =   /0 . ,
[DEFAULT]
                                                   /R ,              [R]
                            ),

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

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

          /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         ,
[function]
                              ♦   COEF=r         ,               [R]
                            ),

          /COMB_C        =_F    ( ♦fonction=        f_c,
[fonction_c]
                              ♦/COEF_R=         R ,                 [R]
                                 /COEF_C        = C   ,                [C]
                            ),

          /MULT      =_F    ( ♦fonction=f         ,
[function]
                            ),

          /REGR_POLYNOMIALE       =_F    ( ♦FONCTION=f         ,
[function]
```

```
                                        ♦DEGRE=n          ,              [I]
                                ),
                        # if COMB or COMB_C or REGR_POLYNOMIALE
                        will ◊LISTE_PARA=lpara          ,
[listr8]

          /COMPOSE      =_F    ( ♦fonc_RESU=f_resu              ,
[function]
                               will ♦fonc_PARA=f_para           ,
[function]
                               ),

          /ASSE         =_F    ( ♦FONCTION=l_f          ,
[l_fonction]
                               “             RIGHT” ◊SURCHARGE=/,
[DEFAULT]
                                                    /  “GAUCHE”,
                               ),

          /EXTRACTION   =_F    ( ♦fonction=f_c          ,
[fonction_c]
                               ♦PARTIE=/                “REEL”,
                                                    /  “IMAG”,
                                                    /  “MODULE”,
                                                    /  “PHASE”,
                               ),

          /PUISSANCE     =_F    ( ♦FONCTION=f           ,
       [function]
                               ◊EXPOSANT=/          N,            [I]
                                                   /1,          [DEFAULT]
                               ),

          /FFT           =_F    ( ♦FONCTION=f           ,
[function]
                               ◊METHODE=/               “PROL_ZERO”,
[DEFAULT]
                                                    /  “TRONCATURE”,
                                                    /  “COMPLET”,
                               ◊    SYME=/              “OUI',
[DEFAULT]
                                                    /  “NON”,
                               ),

          /CORR_ACCE     = _F (   ♦fonction=f           ,
[function]
                               ◊CORR_DEPL=/             “NON”,
[DEFAULT]
                                                    /  “YES”,
                               ),

          /LISS_ENVELOP     = _F (
                               ♦    NAPPE=n          ,                [three-
dimensions function]
                               ◊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]
                              ),


        /    DSP =_F    (
                              ♦FONCTION=sro           ,              [function]
                              ♦AMOR_REDUIT=lam        ,             [l_R]
                               ◊ /FREQ          =lfreq   ,             [l_R]
                                 /LIST_FREQ        =liste_freq ,       [listr8]
                              ♦FREQ_COUP       =frc                    [R]
                              ♦DUREE=tsm                              [R]
                              ♦NORME=r                ,               [R]
                              ◊   FRACT=/0.5                ,
[DEFAULT]
                                                      /fract           [R]


       /SPEC_OSCI     =_F    ( ♦fonction=f             ,
[function]
                                ◊             METHODE=' NIGAM',
[DEFAULT]
                                ◊   AMOR_REDUIT=LAM     ,          [l_R]
                                ◊   /FREQ          =LFRE ,          [l_R]
                                    /   LIST_FREQ=lfreq     ,          [listr8]
                                ◊   NATURE=/            “ACCE”,
[DEFAULT]
                                                          /“QUICKLY”,
                                                          /“DEPL”,
                                                 ◊NATURE_FONC=' ACCE',
[DEFAULT]
                                ♦   NORM          =R  ,             [R]
                              ),

           will ◊NOM_PARA=para                ,                          [KN]
            ◊   NOM_RESU=resu                ,                         [K N ]
           ◊PROL_DROITE=/                    “CONSTANT”,
                                          /   “LINEAIRE”,
                                          /   “EXCLUDED”,
           ◊PROL_GAUCHE=/                       “CONSTANT”,
                                          /   “LINEAIRE”,
                                          /   “EXCLUDED”
                                       ◊INTERPOL=I' LIN',
[l_kn]
                                             I'LOG',
                                          |  “NON”,
           ◊   INTERPOL_FONC=I               “LIN”,                       [l_kn]
                                          |  “LOG”,
                                          |  “NON”,
           ◊   NOM_PARA_FONC=parf               ,                         [KN]
           ◊PROL_DROITE_FONC=/               “CONSTANT”,
                                          /   “LINEAIRE”,
                                          /   “EXCLUDED”,
           ◊PROL_GAUCHE_FONC=/               “CONSTANT”,
                                          /   “LINEAIRE”,
                                          /   “EXCLUDED”,
           ◊INFO=/1                ,
[DEFAULT]
```

```
                                /2 ,
    )
```

So factor key word COMB_Calorsfr = [FONCTION_C],
so factor key word SPEC_OSCIalorsfr = [THREE-DIMENSIONS FUNCTION],
so factor key word ENVELOPPE, FRACTILE, PUISSANCE then Fr is of the same type as the functions in entry,
for all the others factor key word, Fr = [FONCTION].

3 Operands

3.1 Key word DERIVE

/DERIVE =

the function is derived $f(t)$.

♦fonction = F

Name of the function which one wishes to derive.
Does not apply to the concepts of type three-dimensions function.

◊METHODE =

Name of THE METHODE which one wishes to use : the only method available is currently DIFF_CENTREE (by default).

Note:

See key word INTEGRE .

3.2 Key word INTEGRE

/INTEGRE =

One integrates the function $f(t)$.

◊COEF = r

Constant of integration, by default 0.

♦FONCTION = F

Name of the function which one wishes to integrate.
Does not apply to the concepts of type three-dimensions function.

◊METHODE =

Name of THE METHODE which one wishes to use.

Two methods are available: method of the "TRAPEZE" (by default) and method of "SIMPSON".

The integral is exact for the linear functions per pieces as provided as starter. The mistake made compared to the integral of the function which one discretized is in $o(1/n^2)$ for functions of classes C^2 .

The method of Simpson is exact for the polynomials of degree lower or equal to 3. The error is in $o(1/n^4)$ for functions of classes C^4 .

For the unspecified functions, very kicked up a rumpus, like the accelerograms, it is advised to use the method of the trapezoids.
On the other hand, when the function $f(t)$ (before discretization) is sufficiently regular, the method of Simpson is much more precise.

Note:

1) *For INTEGRE as for DERIVE, the NOM_PARA of the produced function is unchanged. On the other hand, the NOM_RESU can be modified in the following cases: for derivative, DEPL becomes QUICKLY, QUICKLY becomes ACCE; for integration, ACCE becomes QUICKLY, QUICKLY becomes DEPL. The user always has the possibility of modifying it by the key word of the same name in CALC_FONCTION.*

2) *Concerning the prolongations, the produced function has, by default of the prolongations* EXCLUDED *on the left and on the right some are those of the starting function. Not to thus expect that a linear prolongation becomes constant in the derived function… There still, the user is Master of his prolongations for the function produced by key keys* PROL_DROITE *and* PROL_GAUCHE.

3.3 Key word INVERSE

/INVERSE =

One reverses the function $f(t)$.

♦FONCTION = F

Name of the function which one wishes to reverse, it is necessary that this one is bijective (strictly increasing or strictly decreasing).

Does not apply to the concepts of type three-dimensions function.

Note:

1) *The labels of the parameters are not reversed! The care is left to the user affect the correct values by key keys* NOM_PARA *and* NOM_RESU. *By default,* the NOM_PARA *is unchanged and* NOM_RESU *is affected with* "TOUTRESU" .
2) *The modes of interpolations are inverted: e.g. (* "LIN" , "LOG" *) becomes (* "LOG" , "LIN" *).*
3) *The prolongations* EXCLUDED *and* LINEAIRE *are unchanged. On the other hand, a prolongation* CONSTANT *is changed into* EXCLU.

3.4 Key word AB

/ABS =

Provides the absolute value of a function or a three-dimensions function.

♦FONCTION = F

Name of the function which one wishes the absolute value.

Note:

1) *Parameters (prolongations, interpolations,* NOM_PARA *and* NOM_RESU*) of the produced function are the same ones as those of the starting function.*
2) *Except for prolongation* LINEAIRE *: systematically changed into* EXCLU *by precaution. Indeed, the linear prolongation on the right of a decreasing function leads for sufficiently large X-coordinates to negative values: responsibility is thus left to the user affect itself* PROL_DROITE=' LINEAIRE' *(and respectively on the left).*

3.5 Key word ENVELOPPE

/ENVELOPPE =

Computation of the envelope of several functions.

This operation is available on operands of natural function or three-dimensions function.

3.5.1 Operand FONCTION

♦FONCTION = F

Lists functions or three-dimensions functions which one seeks the envelope.

3.5.2 Operand CRITERE

◊CRITERE =

/ "SUP"

One seeks the higher envelope.

/ "INF"

One seeks the lower envelope.

Remarks for the search of the envelope:

- *the functions all must be of comparable nature (function or three-dimensions function),*
- *Case of the simple functions: for the prolongations, interpolations, NOM_PARA and NOM_RESU, they are the parameters of the first of the functions in the list which are retained. The support of X-coordinates of the function envelope will be the meeting of the lists of X-coordinates of all the functions.*
- *Case of the three-dimensions functions: parameters (prolongations, interpolations, NOM_PARA, NOM_RESU, NOM_PARA_FONC) must imperatively be identical between the provided three-dimensions functions. The supports of X-coordinates (values of the parameters and X-coordinates of the functions of the three-dimensions functions) are homogenized to be able to calculate the envelope. The produced three-dimensions function will have this discretization for X-coordinates.*

3.6 Key word FRACTILE

/FRACTILE =

Computation of the fractile several functions.

This operation is available on operands of natural function or three-dimensions function.

3.6.1 Operand FONCTION

♦FONCTION = F

Lists functions or three-dimensions functions which one seeks to calculate the fractile.

3.6.2 Operand FRACT

♦FRACT = fract

Value of the quantile to calculating. By default fract = 1, the fractile is then the higher envelope.

3.7 Key word COMB and operand LIST_PARA

/COMB =

real linear Combination several concepts of natural function or three-dimensions function.

♦FONCTION = F

Name of the function to be combined.

♦COEF = R

Value of the coefficient.

◊LISTE_PARA = will lpara

List of the values of the parameters for which the combination of the functions will be discretized. If this key word is not indicated, a list by default is built by taking the union of the lists of the values of the parameters of each function.

Caution:

It is not a key word of the key word factor COMB .

Remarks for the combination:

See the remarks for key word ENVELOPPE

3.8 Key word COMB_C and operand LIST_PARA

linear /COMB_C

= Combination complexes several concepts of nature fonction_c.

♦FONCTION =f_c

Name of the function to be combined. It can be with complex or real values.

/COEF_R = R,
/COEF_C = C,

Value of the multiplying coefficient, either in real form R, or in form complexes C.

◊LISTE_PARA = will lpara

List of the values of the parameters for which the combination of functions will be discretized. If this key word is not indicated, a list by default is built by taking the union of the lists of the values of the parameters of each function.

Remarks for the combination:

See the remarks for key word ENVELOPPE

3.9 Key word MULT and operand LIST_PARA

/MULT

Produces concepts of comparable nature function, fonction_c or three-dimensions function.

♦FONCTION =f

Nom of the function to be multiplied with the others.

LIST_PARA makes it possible result to discretize the function as for COMB.

3.10 Key word COMPOSE

Key word factor allowing to calculate the made up one of two functions $F(G(t))$.
Does not apply to the concepts of type three-dimensions function.

/COMPOSE =

♦FONC_RESU = f_resu

Function f_resu (X)

♦FONC_PARA = will f_para

Function will f_para (T)

One checks that the NOM_PARA of f_resu corresponds to the NOM_RESU of will f_para.

3.11 Key word ASSE

/ASSE =

Key word factor allowing to create a real function by concaténant two tabulated real functions. Does not apply to the concepts of type three-dimensions function.

3.11.1 Operand FONCTION

♦FONCTION = l_f

Functions with concatenate. Two functions exactly are expected.

3.11.2 Operand SURCHARGE

◊SURCHARGE = "RIGHT",
/ "GAUCHE",

the points of discretization of the function created are those of all the two functions, modulo the effects of overload.
If the fields of definition of the functions overlap, one of the functions imposes its points on the zone of covering and for the prolongations :

SURCHARGE =/"RIGHT" : it is the function which has largest xmax which is chosen,
SURCHARGE =/"GAUCHE" : it is the function which has smallest xmin which is selected.

3.11.3 Checks

One checks that all the functions have the same NOM_PARA, as well as the same interpolations.

3.12 Key word EXTRACTION

/EXTRACTION =

Key word factor allowing to build from a complex function (standard fonct_c), a real function representative either the real part, or the imaginary part, or the modulus, or the phase of the complex function.

3.12.1 Operand FONCTION

complex ♦FONCTION =

f_c Function.

3.12.2 Operand PARTIE

♦PARTIE =

/ "REEL" : extraction of the real part of f_c,
/ "IMAG" : extraction of the imaginary part of f_c,
/ "MODULE" : extraction of the modulus of f_c,
/ "PHASE" : extraction of the phase (in degree) of f_c.

3.13 Key word PUISSANCE

This key word makes it possible to build the nth power of a function or a set of functions provided in the form of a three-dimensions function.

♦FONCTION = F

Name of the function F concerned (standard function or three-dimensions function).

♦ EXPOSANT = N
the function result calculated will be $x \to f(x)^n$. By default $n=1$.

3.14 Key word REGR_POLYNOMIALE

This key word calculates the polynomial regression of a function by the method of the least squares (by means of the function polyfit of numpy).

♦FONCTION = F

Name of the function F concerned (standard function).

♦ DEGRE = N
Degree of the required polynomial.

One can use key word LIST_PARA for tabuler the calculated polynomial. If not, it is tabulated on the list of the X-coordinates of the function F.

3.15 Key word FFT

/FFT =

One calculates the transform of Fourier direct or opposite of a function (of which by algorithm FFT).

♦FONCTION = F

Name of the function on which the operation is carried out.
If the NOM_PARA of the function is INST, then the FFT direct is calculated.
If the NOM_PARA of the function is FREQ, then the opposite FFT is calculated.
Does not apply to the concepts of type three-dimensions function.

♦ METHODE =

algorithm FFT is faster for the samples of which the length is a power of 2.
Method "PROL_ZERO" (by default) proposes to prolong the input signal with zeros until having a nombre total of sample which is the first power of 2 whose value is higher than the initial number of samples.
Method "TRONCATURE" will consider only the first samples of which the nombre total is the greatest power of two whose value is lower than the initial number of sample.
For example, on a signal of 601 values, method "PROL_ZERO" will supplement the signal to have 1024 samples, whereas method "TRONCATURE" will consider only the first 512 times.
If the entry signal has a number of sample which is a power of two, the two methods are obviously equivalent: one takes into account the signal without modifying it.
Method "COMPLET" makes it possible to take into account the totality of the input signal, some is the number of samples.
Nota bene: in the case of a sample length N, of which time step would be dt, the sampling rate of the FFT is $1/(N.dt)$. On the other hand, the last frequency for which the discrete transform is calculated is not $1/dt$, but $(N-1)/(N.dt)$.

♦ SYME =

Key word which applies only for the opposite transform of Fourier.
If the complete spectrum would be provided, then the opposite transform is calculated directly by means of SYME = "OUI'. Methods "TRONCATURE" and "PROL_ZERO" are then not active.
If the spectrum (complex) provided in entry of the FFT reverses does not contain the folded up part (partners with the negative frequencies of the spectrum), one can nevertheless consider a signal temporal having the same spectral contents on the part associated with the positive frequencies. If one notes X_k k the ième sample of the transform of Fourier of a sample length N,

then one has $X_k = X^*_{(N-k)}$, where $()^*$ corresponds to the combined complex. This information can be exploited to rebuild a temporal signal by knowing only half of the spectrum. This operation is carried out when one chooses SYME = "NON". The temporal signal is then rebuilt to obtain a temporal sample even length. In theory, to rebuild a temporal signal length $2 \times M$, the spectrum must check certain conditions:

1. The spectrum must be length $M+1$,
2. the first point of the spectrum must be real,
3. the last point of the spectrum must be real.

If these conditions are not checked, then one builds an approximate spectrum odd length checking these conditions. If the initial spectrum is even length, the last point is then rebuilt by carrying out a prolongation by interpolation of the initial spectrum. This reconstruction can introduce a light skew when the spectral contents of the sample are very significant on the last points of the spectrum.
Methods "TRONCATURE" and "PROL_ZERO" are still available for the opposite FFT. Attention, however, with the use of method "TRONCATURE". If the number of truncated point is significant, then the results can be very appreciably different.

3.16 Key word CORR_ACCE

/CORR_ACCE =

Key word factor allowing to correct an accelerogram measured for computation of the seismic response of a system.
One removes the drift of the signal, calculated by linear lissage within the meaning of the least squares on the totality of the signal, in order to make the accelerogram more realistic. The drift corresponding relative velocity is also removed.

One turns over in output the corrected accelerogram.

3.16.1 Operand FONCTION

♦FONCTION = F

measured real Accelerogram.
Does not apply to the concepts of type three-dimensions function.

3.16.2 Operand CORR_DEPL

◊CORR_DEPL =

/ "NON"

One does not correct the drift of relative displacement, it is the default value.

/ "OUI'

One removes also the drift of relative displacement. This option is to be used with precaution, because one does not know a priori the value of final displacement after the seisme.

3.17 Key word LISS_ENVELOP

the data of origin make up of a three-dimensions function of spectrums SRO gross definite on a large number of points for a level of bottom given.
The first stage consists, for each spectrum, with widening in frequency (shift on the left and on the right) followed by a reduction amongst point of definition. These operations carried out, one makes sure of the character wraps spectrum smoothed compared to the initial spectrum. A this stage, each spectrum has its own base of frequency.
The second stage consists in homogenizing the base of frequency of all the spectrums of the three-dimensions function while making sure of nonthe covering of the spectrums between them.

♦NAPPE = N

Name of the three-dimensions function of entry formed by the rough spectrums associated with each level of damping.

◊FREQ_MIN and FREQ_MAX

Beach of definition in frequency of the smoothed spectrum.
The frequencies mentioned under FREQ_MIN and FREQ_MAX must be selected among the frequencies of discretization of the rough spectrum.
By default, one considers the complete spectrum.

◊ELARG

widening relates to the group of the spectrum,
It is given expressed as a percentage and is worth 0.1 (10%) by default.
For each Fi frequency of the rough spectrum, one defines two new values of frequencies such as:

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

The parameters τ_g and τ_d represent the amplitude of widening in frequency.

The values of the offset frequencies F^- and F^+ do not correspond to the values F_i of the list of definition of the rough spectrum. One defines thus F_j and F_k such as:

- F_j : value belonging to the list, immediately below or equalizes with F^-
- F_k : value belonging to the list, immediately below or equalizes with F^+ .

For each frequency F_i, two points of coordinates (F_j, γ_i) and (F_k, γ_i) are defined where γ_i acceleration with the frequency represents F_i. Two new spectrums resulting from the shift of the rough spectrum on the axis of the frequencies are thus built.

◊TOLE_LISS

bearing expressed as a percentage Criterion on the elimination of the points during lissage. This tolerance is fixed at 0.25 time the value by default.
The lissage is carried out on the envelope of the spectrums rough, shifted on the right and on the left.

An example of application is proposed in the case test ZZZZ100E.

3.18 Key word SPEC_OSCI

/SPEC_OSCI =

Calculates the oscillator spectrum of an accelerogram, function of natural function [R4.05.03].

The oscillator spectrum is computable only on the functions of NOM_RESU = "ACCE" and of NOM_PARA = "INST".

For all i and all j one considers q_j^i the solution of the differential equation:

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

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

the product concept fr is a three-dimensions function (function with two variables) made up by the functions $(fr_i, \ldots, fr_j, \ldots)$ with fr_j function defined in the points ω_i with:

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

By default for the computation of the oscillator spectrum

- one considers for reduced dampings the values:

  0.020.050.10

- one considers for the frequencies, the 150 values following in Hz ,

  first is with $0.2\,Hz$ and one deduces the following ones by the rule;

2nd	with	the 57ème : by step de	0.05 Hz
58		650.075	Hz
66		790.10	Hz
80		1030.125	Hz
104		1310.25	Hz
132		1370.5	Hz
138		1411	. Hz
142		1501.5	Hz

  the cut-off frequency by default is thus $35{,}5\,Hz$ (the user must check that this value is coherent with the frequential contents of the entry signal, if it is not the case, it is necessary to define a list of adapted frequency).

- the spectrum is normalized according to the value of NORM.

3.18.1 Operand FONCTION

♦FONCTION =f

Nom of the function on which the operation is carried out.
Does not apply to the concepts of type three-dimensions function.

3.18.2 Operand METHODE

◊METHODE =

Name of THE METHODE which one wishes to use : the only method usable currently is "NIGAM" (by default) which is detailed in the document [R5.05.01].

3.18.3 Operand AMOR_REDUIT

◊AMOR_REDUIT = lam

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

Lists reduced dampings : example 0.01,0.05,…

3.18.4 Operands FREQ/LIST_FREQ

~ ◊FREQ = lfre

lfre = $\left(\varphi_1, ..., \varphi_i, ...\right)$. List frequencies.

~ ◊LISTE_FREQ = lfreq

List of the frequencies provided under a concept listr8.

The frequencies must be strictly positive.

3.18.5 Operands NATURE/NATURE_FONC

◊NATURE =

Natural of the quantity of three-dimensions function created by the command CALC_FONCTION.

ACCE	Spectrum of pseudo-acceleration	$\ddot{u}(t)=\omega_i^2 u(t)$
Spectrum	of pseudo-acceleration QUICKLY	$\dot{u}(t)=\omega_i u(t)$
DEPL	Spectrum of pseudo-acceleration	$u(t)$

◊NATURE_FONC = "ACCE"

Nature of the function which is used to build the spectrum. For time only value "ACCE" is available. This key word makes it possible to overload the NOM_RESU of the function specified under key word FONCTION when this one is created by RECU_FONCTION [U4.32.03].

3.18.6 Operand NORMALIZES

♦NORME = R

the oscillator spectrum will be normalized with the value R (value of pseudo-acceleration), this value is recalled in the file of message.

3.19 Key word DSP

/DSP =

Calculates power spectral density (DSP) equivalent to the data of a response sprectrum of oscillator (SRO) with the formula of Vanmarcke, function of natural function [R4.05.03].

The oscillator spectrum is computable only on the functions of NOM_PARA = "FREQ".

For the computation of the DSP from an oscillator spectrum, one considers that

- the SRO expresses the median maximum (fractile 0.5), if not it is necessary to inform another value via key word FRACT.
- the DSP is worth zero for frequencies lower or equal to $1/2\ \pi Hz$.
- the spectrum is normalized according to the value of NORM.

The user must check that the frequential discretization (list of the frequencies) is sufficient compared to the frequential contents of the signals to modelling. It is also advisable to check equivalence between the DSP and the SRO given by making pullings or by determining values of maximum response of an oscillator with POST_DYNA_ALEA. The operator calculates power spectral density (DSP) equivalent to the data of a response sprectrum of oscillator (SRO) with the formula of Vanmarcke. One does not carry out an iteration to optimize did it.
An example of application is proposed in the case test ZZZZ100D.

3.19.1 Operand FONCTION

♦FONCTION =sro

Name of the function defining the SRO.

3.19.2 Operands AMOR_REDUIT, FREQ, LIST_FREQ

Key words identical to SPEC_OSCI.

3.19.3 Operands FREQ_COUP

◊FREQ_COUP = frc

the cut-off frequency: one determines the DSP until this frequency. The SRO is prolonged (constant value corresponding to the ZPA) up to this value so necessary.

3.19.4 Operand DUREE

♦DUREE = tsm

Lasted of the strong phase of the seisme. The seismic signal is regarded as steady over this period. This value is necessary to evaluate the factor of peak which intervenes in the formula of Vanmarcke.

3.19.5 Operand NORMALIZES

♦NORME = R

One considers oscillator spectrums normalized with the value R (value of pseudo-acceleration). In general, the SRO are given in G, it is thus necessary to inform NORM = 9.81.

3.20 Attributes of the concept function in output

3.20.1 Default values

By default the attributes of the concept function in output of the command CALC_FONCTION are for the various options (cf commands DEFI_FONCTION [U4.31.02] and DEFI_NAPPE [U4.31.03]).

- Option DERIVE:

Interpolation : data by the function as starter

left Prolongation: EXCLU

right Prolongation: EXCLU

NOM_PARA = "INST" (example) given by the function in entry

NOM_RESU = "QUICKLY" (example) given by the function in entry

- Option INTEGRE:

Even rules that for DERIVE

- Options COMB / COMB_C :

Attributes of the first combined function.

- Option SPEC_OSCI: result is a three-dimensions function

the attributes of the three-dimensions function:

NOM_PARA = "AMOR"

NOM_RESU = "DEPL" or "QUICKLY" or "ACCE"

Interpolation: "LOG"

left Prolongation: "EXCLUDED"

right Prolongation: "EXCLUDED"

the attributes from each function:

NOM_PARA = "FREQ"

Interpolation : "LOG"

left Prolongation: "EXCLUDED"

right Prolongation: "CONSTANT"

- Option ENVELOPPE:

Attributes of the first function given.

- Option FFT :

NOM_PARA = FREQ if NOM_PARA of the function is INST
If not it is the reverse

- Option COMPOSE :

NOM_PARA : that of function FONC_PARA
NOM_RESU : that of function FONC_RESU
INTERPOL : that of function FONC_RESU
Prolongation: that of function FONC_RESU

- Option EXTRACTION :

Attributes identical to those of the function given in entry

- Option ASSE :

NOM_PARA : that of functions
NOM_RESU : that of functions
INTERPOL : linear
Prolongation: "EXCLUDED"

3.20.2 Overload from the attributes

the user can by means of overload the attributes given by default the following key keys:

3.20.2.1 Operand NOM_PARA

◊NOM_PARA = para

It indicates the name of the parameter (variable or X-coordinate) of the function or the three-dimensions function. The values currently authorized for para are:

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

3.20.2.2 Operand NOM_RESU

◊NOM_RESU = resu

It makes it possible to document, the function created by giving a name (8 characters) to the function. Except exception (cf [§3.1], [§3.2], [§3.5]), this name is not tested.

3.20.2.3 Operand INTERPOL

◊INTERPOL

When the product concept is a three-dimensions function, INTERPOL defines the type of interpolation between two consecutive values of the parameter of the three-dimensions function and between two functions (once those evaluated). Even meaning that in DEFI_NAPPE.
When the product concept is a function (real or complex), it defines the type of interpolation for the X-coordinates and the ordered of the function. One waits up to two values. If it only one value is provided, it is used for the X-coordinates and the ordered. Even meaning that in DEFI_FONCTION.

3.20.2.4 Operands PROL_DROITE/ PROL_GAUCHE

◊PROL_DROITE and PROL_GAUCHE

They define the type of prolongation on the right (respectively on the left) of the field of definition of the variable:

- "CONSTANT" for a prolongation with the last (or the first) value of the function,
- "LINEAIRE" for a prolongation along the first definite segment (PROL_GAUCHE) or last definite segment (PROL_DROITE),
- "EXCLUDED" if the extrapolation of the values apart from the field of definition of the parameter is prohibited.

3.20.2.5 Operands NOM_PARA_FONC / INTERPOL_FONC / PROL_DROITE_FONC / PROL_GAUCHE_FONC

These keys key make it possible to modify the attributes of the three-dimensions function and apply to the parameters of the functions of this one. They have the same meaning as the keywords without suffix FONC.

- NOM_PARA_FONC is the name of the parameter of the functions (as in DEFI_NAPPE).
- INTERPOL_FONC is the type of interpolation for the X-coordinates and Y-coordinates of the functions of three-dimensions function (identical to the key word INTERPOL of factor key word the DEFI_FONCTION of DEFI_NAPPE).
- PROL_GAUCHE_FONC/PROL_DROITE_FONC define the prolongations of the functions (identical to key words PROL_GAUCHE/PROL_DROITE of factor key word the DEFI_FONCTION of DEFI_NAPPE).

3.21 Operand INFO

◊INFO

If INFO=2, one prints function (IMPR_FONCTION format TABLEAU) in the message file .

4 Computation

4.1 examples of an envelope

the command file which follows:

```
DEPI=2. * pi
PAS0=DEPI/200.
LI1=DEFI_LISTE_REEL (  DEBUT=0.,
                     INTERVALLE=_F (  JUSQU_A = DEPI, NOT = PAS0))

COa = FORMULA (NOM_PARA=' INST', VALE=' cos (INST) ")
SIa = FORMULA (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',
                            TITER=' FONCTION COSINUS'   )

IF  = CALC_FONC_INTERP (     FONCTION=SIa,    LIST_PARA=LI1,
                            NOM_PARA=' INST',
                            NOM_RESU=' DEPLACEMENT',
                            PROL_GAUCHE=' EXCLU',
                            PROL_DROITE=' CONSTANT',
                            INTERPOL=' LIN',
                            TITER=' FONCTION SINE "    )

ENV1=CALC_FONCTION (ENVELOPPE=_F (  FONCTION = (IF,  CO,),
                                  CRITERE = "SUP") )
```

4.2 Computation of derivative of the function `if`

the commands which follow

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

      inst1 = 20. * not

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

produce on file "RESULTAT" :

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

4.3 Concatenation of two functions

```
      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 = "RIGHT")
                        )

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

the values of the function dfc3 are:

```
   x=0    .       4.     5.     7.     8.
   y=      10.14.25.27                   .28      .
```

The values of the function dfc4 are:

```
   x=0    .       4.     6.     7.     8.
   y=      10.14.16.27                   .28      .
```

4.4 Composition of two functions

```
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    )
                         )
```

the values of the function comp1 are:

```
inst=0.0.1                  0.2.0.3      0.40.50.60                 .
70.80.91.0
F=0.5.12.514.10.9.8.3.0.80.40
                         .
```