

Operator DEFI_LIST_FREQ

1 Goal

To create a list of realities while possibly refining around values of frequencies informed by the user.

Product a structure of data of the type `listr8`.

2 Syntax

```

Lr      [listr8] = DEFI_LIST_FREQ
      (
          ♦ / VALE=      lr8      ,                [l_R]
          / ♦ DEBUT=     debu     ,                [R]
          ♦ INTERVALLE= (_F ( ♦ JUSQU_A =      r1, [R]
                              ♦ / NUMBER =     n1, [I]
                              / NOT =          r2, [R]
                              ), ),
          ◇ RAFFINEMENT= (_F (
                              ♦ LIST_RAFFINE =   lr8      ,                [l_R]
                              ◇ NB_POINTS =     / 5      ,                [DEFECT]
                              / Pt              ,                [I]
                              ◇ PAS_MINI =      / 0,001   ,                [DEFECT]
                              / not           ,                [R]
                              ◇ CRITERION =     / 'RELATIVE' ,
                              / 'ABSOLUTE'     ,
                              / 'LARGEUR_3DB' ,
                              # Keyword only associated with the criteria 'RELATIVE' and 'ABSOLUTE' :
                              ◇ DISPERSION =     / 0.01   ,                [DEFECT]
                              / disp           ,                [R]
                              # Keyword only associated with the criterion 'LARGEUR_3DB' :
                              ♦ / AMOR_REDUIT =   lr8_amor, [l_R]
                              / LIST_AMOR      =   l_amor , [listr8]
                              ◇ INFORMATION =     / 1     ,
                              [DEFECT]
                              / 2     ,
                              ◇ TITLE =         title ,                [l_Kn]
          )

```

3 Operands

3.1 Operand `VALE`

`VALE = lr8`

List of "basic" realities which will automatically be part of the structure of data `listr8` result.
This list can be built starting from a list Python.

3.2 Operand `BEGINNING`

◆ `BEGINNING = debu`

The first reality of the list of "basic" realities which one wants to build.

3.3 Operand `INTERVAL`

◆ `INTERVAL =`

◆ `JUSQU_A = r1`

End of the interval which one will cut out with a constant step.

◆ `/ NUMBER = n1`

Many steps which one wants in the interval which ends in `r1`.

`/ NOT = r2`

Pas de division interval.

Notice :

All the values required in the basic list are automatically in the final list at exit of `DEFI_LIST_FREQ`.

When the keyword is used `NOT` it may be that the number of calculated step is not rigorously whole. One "will then adapt" the last interval to fall down exactly on the end value (`JUSQU_A`).

3.4 Keyword `REFINEMENT`

3.4.1 Operand `LIST_RAFFINE`

◆ `LIST_RAFFINE = lr8`

List of the frequencies around which one wishes to refine.

This list can be built starting from a list Python.

This keyword is particularly interesting in the case of a harmonic analysis of a structure (operator `DYNA_VIBRA` [U4.53.03]), because it will make it possible to calculate the harmonic answer around the Eigen frequencies of the structure and thus to recover the local extrema of them. It is possible to directly recover the Eigen frequencies in a structure of data of the type `mode_XXX` exit of a modal calculation carried out for example with the order `CALC_MODES`. It is enough for that to use the function `LIST_VARI_ACCES ()`.

It is pointed out that it is then essential to carry out the code in mode `PAR_LOT='NON'` (keyword of the order `BEGINNING` or `CONTINUATION`. For more details to see documentation [U1.03.02]).

Example:

```
MODES=CALC_MODES (MATR_RIGI=MATASSR,  
                 MATR_MASS=MATASSM,
```

```
OPTION = 'ADJUSTS',  
CALC_FREQ=_F (FREQ= (5. , 10. , 15. , 20. ,  
24. , 27.))  
)
```

```
list_freq = MODES.LISTE_VARI_ACCES () ['FREQ']
```

`list_freq` is thus a list python containing the list of the Eigen frequencies present in the result `MODES`.

3.4.2 Operand PAS_MINI

```
◇ PAS_MINI = / 0,001 , [R]  
/ not , [R]
```

If the difference between two values is lower than `PAS_MINI` then one of them is removed.

Exception: if the two values concerned belong one to the list given under `LIST_RAFFINE` and the other with the basic list, then they are kept both.

3.4.3 Operand DISPERSION

```
◇ DISPERSION = / 0.01 , [DEFECT]  
/ disp , [R]
```

`DISPERSION` is the width of the interval surrounding each frequency of `LIST_RAFFINE` where one wants to refine.

3.4.4 Operand CRITERION

```
◇ CRITERION = / 'RELATIVE' , [DEFECT]  
/ 'ABSOLUTE' ,  
/ 'LARGEUR_3DB' ,
```

If `CRITERION = 'RELATIVE' or 'ABSOLUTE' :`

```
◇ DISPERSION = / 0.01 , [DEFECT]  
/ disp , [R]
```

The width of the interval df surrounding each frequency $freq_i$ of `LIST_RAFFINE` where one wants to refine is worth:

- If `CRITERE=' RELATIF' :`
 $df = disp \cdot freq_i$
- If `CRITERE=' ABSOLU' :`
 $df = disp$

If `CRITERE=' LARGEUR_3DB' :`

```
◇ / AMOR_REDUIT = lr8_amor
```

List of reduced depreciation ($\eta_1 , \eta_2 , \dots , \eta_n$ expressed as a percentage) corresponding to each clean mode of the system in the form of list of realities.

```
/ LIST_AMOR = l_amor
```

Name of the concept of the type `listr8` containing the list of reduced depreciation ($\eta_1, \eta_2, \dots, \eta_n$ expressed as a percentage).

Notice :

If the number of reduced depreciation given is lower than the number of basic vectors used in the modal base, depreciation of the additional vectors is taken equal to the last damping of the list.

If damping η_i is nonnull, the frequency $freq_i$ used to calculate Llength of the interval has `df` is the frequency $freq_i$ of `LIST_RAFFINE` shifted at the frequency of resonance of a forced system: $freq_i = freq_i \sqrt{1 - 2\eta_i^2}$.

The length of the interval `df` surrounding each frequency $freq_i$ where one wants to refine is worth:

- If $\eta_i \neq 0$:

$$df = 2\eta_i freq_i$$

- If $\eta_i = 0$:

$$df = 0.01 freq_i$$

3.4.5 Operand NB_POINTS

◇ NB_POINTS = / 5 , [DEFECT]
/ Pt , [I]

`NB_POINTS` is the number of points which one wants to add around the frequencies of the list `LIST_RAFFINE`.

The added points are uniformly distributed in the interval `df`. Thus if `NB_POINTS` is an odd number the frequencies of `LIST_RAFFINE` will be in the final list.

If `CRITERE='LARGEUR_3DB'` and `NB_POINTS` is an even number, the frequency is kept $freq_i$ of `LIST_RAFFINE` and the frequency of resonance is added of a forced system such as: $freq_i = freq_i \sqrt{1 - 2\eta_i^2}$.

3.5 Operand INFORMATION

◇ INFORMATION = I

Indicate the level of impression of the results of the operator.

- 1: no impression,
- 2: impression of the list of realities created

3.6 Operand TITLE

◇ TITLE = title

Title which the user wants to give to his list of realities.

4 Examples

The cas-test sdd21b presents an example of use of DEFI_LIST_FREQ.

Example 1:

Let us imagine that one wants to create the basic list:

1. 3. 5. 10. 15.

who is such as the step is: 2. of 1. with 5.
 5. of 5. with 15.

And that one wants to refine around frequency 3.5 with the criterion 'ABSOLUTE'.

One can write:

```
Lr = DEFI_LIST_FREQ (BEGINNING = 1. ,
                    INTERVAL = (_F (JUSQU_A= 5. , NOMBRE= 2, ),
                              _F (JUSQU_A= 15. , PAS= 5, )),
                    REFINEMENT= (_F (LIST_RAFFINE= 3.5,
                                      PAS_MINI= 0,001,
                                      NB_POINTS= 5,
                                      CRITERE=' ABSOLU',
                                      DISPERSION=0.02,))
                    )
```

A list then is obtained listr8 containing the values:

[1. , 3. , 3.49,3.495,3.5,3.505,3.51,5. , 10. , 15.]

Example 2:

By using the criterion 'RELATIVE' for refinement one can write :

```
Lr = DEFI_LIST_FREQ (VALE = [1. , 3. , 5. , 10. , 15.] ,
                    REFINEMENT = (_F (LIST_RAFFINE= 3.5,
                                      PAS_MINI= 0,001,
                                      NB_POINTS= 6,
                                      CRITERE=' RELATIF',
                                      DISPERSION=0.03,))
                    )
```

A list then is obtained listr8 containing the values:

[1. , 3. , 3.4475,3.4685,3.4895,3.5105,3.5315,3.5525,5. , 10. , 15.]