Opérateur DEFI_LIST_FREQ

1 But

Créer une liste de réels en raffinant éventuellement autour de valeurs de fréquences renseignés par l'utilisateur.

Produit une structure de données de type `listr8`.

2 Syntaxe

```
lr     [listr8] = DEFI_LIST_FREQ

       (          ♦  / VALE=     lr8    ,                  [l_R]

                    /  ♦  DEBUT=    debu   ,                  [R]

                       ♦  INTERVALLE=  (_F(  ♦  JUSQU_A =    r1, [R]
                                             ♦  /  NOMBRE =  n1, [I]
                                                /  PAS =     r2, [R]
                                          ),),

                  ◊  RAFFINEMENT= (_F(

                                  ♦  LIST_RAFFINE =  lr8    ,        [l_R]
                                  ◊  NB_POINTS =  /  5      ,        [DEFAUT]
                                                  /  pt     ,        [I]
                                  ◊  PAS_MINI =   /  0.001  ,        [DEFAUT]
                                                  /  pas    ,        [R]

                                  ◊  CRITERE = /  'RELATIF'    ,     [DEFAUT]
                                               /  'ABSOLU'     ,
                                               /  'LARGEUR_3DB',

                  # Mot clé associé uniquement aux critères 'RELATIF' et 'ABSOLU' :
                                  ◊  DISPERSION = /  0.01   ,        [DEFAUT]
                                                  /  disp   ,        [R]

                  # Mot clé associé uniquement au critère 'LARGEUR_3DB' :
                                  ♦  /  AMOR_REDUIT  =  lr8_amor, [l_R]
                                     /  LIST_AMOR    =  l_amor ,  [listr8]


                    ◊  INFO   =  /  1  ,                          [DEFAUT]
                                 /  2  ,

                    ◊  TITRE  =  titre ,                          [l_Kn]

       )
```

3 Opérandes

3.1 Opérande VALE

VALE = lr8

Liste des réels « de base » qui feront automatiquement partie de la structure de données listr8 résultat.
Cette liste peut être construite à partir d'une liste Python.

3.2 Opérande DEBUT

♦ DEBUT = debu

Premier réel de la liste de réels « de base » que l'on veut construire.

3.3 Opérande INTERVALLE

♦ INTERVALLE =

♦ JUSQU_A = r1

Extrémité de l'intervalle que l'on va découper avec un pas constant.

♦ / NOMBRE = n1

Nombre de pas que l'on veut dans l'intervalle qui se termine par r1.

/ PAS = r2

Pas de découpage de l'intervalle.

Remarque :

Toutes les valeurs demandées dans la liste de base sont automatiquement dans la liste finale en sortie de DEFI_LIST_FREQ.
Lorsqu'on utilise le mot clé PAS il se peut que le nombre de pas calculé ne soit pas rigoureusement entier. On "adaptera" alors le dernier intervalle pour retomber exactement sur la valeur finale (JUSQU_A).

3.4 Mot clé RAFFINEMENT

3.4.1 Opérande LIST_RAFFINE

♦ LIST_RAFFINE = lr8

Liste des fréquences autour desquelles on souhaite raffiner.
Cette liste peut être construite à partir d'une liste Python.
Cet mot-clé est particulièrement intéressant dans le cas d'une analyse harmonique d'une structure (opérateur DYNA_VIBRA [U4.53.03]), car il permettra de calculer la réponse harmonique autour des fréquences propres de la structure et donc d'en récupérer les extrema locaux. Il est possible de récupérer directement les fréquences propres dans une structure de données de type mode_XXX issue d'un calcul modal réalisé par exemple avec la commande CALC_MODES. Il suffit pour cela d'utiliser la fonction LIST_VARI_ACCES().
On rappelle qu'il est alors indispensable d'exécuter le code en mode PAR_LOT='NON' (mot clé de la commande DEBUT ou POURSUITE. Pour plus de détails voir la documentation [U1.03.02]).

Exemple :

```
MODES=CALC_MODES( MATR_RIGI=MATASSR,
                  MATR_MASS=MATASSM,
                  OPTION = 'AJUSTE',
                  CALC_FREQ=_F( FREQ= (5.,10.,15.,20.,24.,27.))
                 )

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

list_freq est ainsi une liste python contenant la liste des fréquences propres présentes dans le résultat MODES.

3.4.2 Opérande PAS_MINI

```
◊  PAS_MINI =   /  0.001  ,            [R]
                /  pas    ,            [R]
```

Si l'écart entre deux valeurs est inférieur à PAS_MINI alors l'une d'elles est supprimée.

Exception : si les deux valeurs concernées appartiennent l'une à la liste donnée sous LIST_RAFFINE et l'autre à la liste de base, alors elles sont gardées toutes les deux.

3.4.3 Opérande DISPERSION

```
◊  DISPERSION = /  0.01   ,            [DEFAUT]
                /  disp   ,            [R]
```

DISPERSION est la largeur de l'intervalle entourant chaque fréquence de LIST_RAFFINE où on veut raffiner.

3.4.4 Opérande CRITERE

```
◊  CRITERE = /  'RELATIF'    ,         [DEFAUT]
             /  'ABSOLU'     ,
             /  'LARGEUR_3DB',
```

Si CRITERE = 'RELATIF' ou 'ABSOLU' :

```
◊  DISPERSION = /  0.01   ,            [DEFAUT]
                /  disp   ,            [R]
```

La largeur de l'intervalle df entourant chaque fréquence $freq_i$ de LIST_RAFFINE où on veut raffiner vaut :

- Si CRITERE='RELATIF' :
$$df = disp \,.\, freq_i$$
- Si CRITERE='ABSOLU' :
$$df = disp$$

Si CRITERE='LARGEUR_3DB' :

```
♦  /  AMOR_REDUIT  =  lr8_amor
```

Liste des amortissements réduits (η_1 , η_2 , ... , η_n en pourcentage) correspondants à chaque mode propre du système sous forme de liste de réels.

```
   /  LIST_AMOR =  l_amor
```

Nom du concept de type listr8 contenant la liste des amortissements réduits (η_1 , η_2 , ... , η_n en pourcentage).

Remarque :

Si le nombre d'amortissements réduits donnés est inférieur au nombre de vecteurs de base utilisés dans la base modale, les amortissements des vecteurs supplémentaires sont pris égaux au dernier amortissement de la liste.

Si l'amortissement η_i est non nul, la fréquence $freq_i$ utilisée pour calculer la longueur de l'intervalle df est la fréquence $freq_i$ de LIST_RAFFINE décalée à la fréquence de résonance d'un système forcé : $freq_i = freq_i \sqrt{(1-2\eta_i^2)}$.

La longueur de l'intervalle df entourant chaque fréquence $freq_i$ où on veut raffiner vaut :

- Si $\eta_i \neq 0$:

$$df = 2\,\eta_i\, freq_i$$

- Si $\eta_i \neq 0$:

$$df = 0.01\, freq_i$$

3.4.5 Opérande NB_POINTS

◊ NB_POINTS = / 5 , [DEFAUT]
/ pt , [I]

NB_POINTS est le nombre de points que l'on veut ajouter autour des fréquences de la liste LIST_RAFFINE.

Les points ajoutés sont uniformément répartis dans l'intervalle df . Donc si NB_POINTS est un nombre impair les fréquences de LIST_RAFFINE seront dans la liste finale.

Si CRITERE='LARGEUR_3DB' et NB_POINTS est un nombre pair, on garde la fréquence $freq_i$ de LIST_RAFFINE et on ajoute la fréquence de résonance d'un système forcé tel que : $freq_i = freq_i \sqrt{(1-2\eta_i^2)}$.

3.5 Opérande INFO

◊ INFO = i

Indique le niveau d'impression des résultats de l'opérateur.

1 : aucune impression,
2 : impression de la liste de réels créée

3.6 Opérande TITRE

◊ TITRE = titre

Titre que l'utilisateur veut donner à sa liste de réels.

4 Exemples

Le cas-test sdld21b présente un exemple d'utilisation de DEFI_LIST_FREQ.

Exemple 1 :

Imaginons que l'on veuille créer la liste de base :

```
1. 3. 5. 10. 15.
```

qui est telle que le pas soit : 2. de 1. à 5.
5. de 5. à 15.

Et que l'on veuille raffiner autour de la fréquence 3.5 avec le critère 'ABSOLU'.

On peut écrire :

```
lr   = DEFI_LIST_FREQ   (DEBUT       =  1.,
                         INTERVALLE   =( _F  (JUSQU_A= 5. , NOMBRE= 2, ),
                                        _F  (JUSQU_A= 15., PAS= 5, ),),
                         RAFFINEMENT  =( _F  (LIST_RAFFINE= 3.5,
                                              PAS_MINI= 0.001,
                                              NB_POINTS= 5,
                                              CRITERE='ABSOLU',
                                              DISPERSION=0.02,),
                                       )
                        )
```

On obtient alors une liste listr8 contenant les valeurs :
[1., 3., 3.49, 3.495, 3.5, 3.505, 3.51, 5., 10., 15.]

Exemple 2 :

En utilisant le critère 'RELATIF' pour le raffinement on peut écrire :

```
lr   = DEFI_LIST_FREQ   ( VALE =  [1., 3., 5., 10., 15.] ,
                          RAFFINEMENT =( _F  (LIST_RAFFINE= 3.5,
                                              PAS_MINI= 0.001,
                                              NB_POINTS= 6,
                                              CRITERE='RELATIF',
                                              DISPERSION=0.03,),
                                       )
                        )
```

On obtient alors une liste listr8 contenant les valeurs :
[1., 3., 3.4475, 3.4685, 3.4895, 3.5105, 3.5315, 3.5525, 5., 10., 15.]