Operator COMB_MATR_ASSE

1 Goal

To combine linearly, with real or complex coefficients, of the concepts of the type matr_asse_*.

This operator also allows to carry out linear combinations by considering only the real or imaginary part of a matrix with complex coefficients (extraction of the real or complex part of a matrix).

All concepts of the type matr_asse_* to combine, must divide **same classification**, i.e. the two matrices will have been assembled by the operator ASSE_MATRICE with the same concept argument for the keyword NUME_DDL (cf [U4.61.11]).

Product a structure of data of the type matr_asse_*.

2 Syntax

```
cmass   [matr_asse_*] = COMB_MATR_ASSE    (
   ◊    MATR_ASSE = m ,                          /    [matr_asse_DEPL_R]
                                                 /    [matr_asse_TEMP_R]
                                                 /    [matr_asse_PRES_R]
                                                 /    [matr_asse_DEPL_C]
                                                 /    [matr_asse_TEMP_C]
                                                 /    [matr_asse_PRES_C]
                                                 /    [matr_asse_GENE_R]
                                                 /    [matr_asse_GENE_C]

   ♦    /    COMB_R =  _F (
                ◊    PART =    /    'REAL',
                                /    'IMAG',

                ♦    MATR_ASSE = m ,    /    [matr_asse_DEPL_R]
                                          /    [matr_asse_TEMP_R]
                                          /    [matr_asse_PRES_R]
                                          /    [matr_asse_DEPL_C]
                                          /    [matr_asse_TEMP_C]
                                          /    [matr_asse_PRES_C]
                                          /    [matr_asse_GENE_R]
                                          /    [matr_asse_GENE_C]

                ♦    COEF_R = R    ,              [R]
                          ),
        /    COMB_C =  _F (
                ♦    MATR_ASSE = m ,    /    [matr_asse_DEPL_R]
                                          /    [matr_asse_TEMP_R]
                                          /    [matr_asse_DEPL_C]
                                          /    [matr_asse_TEMP_C]
                                          /    [matr_asse_PRES_R]
                                          /    [matr_asse_PRES_C]
                                          /    [matr_asse_GENE_R]
                                          /    [matr_asse_GENE_C]

                ♦    /    COEF_R = R    ,         [R]
                     /    COEF_C = C    ,         [C]
                          ),
        /    CALC_AMOR_GENE = _F (
                ♦    /    AMOR_REDUIT   = lr8,    [l_R]
                     /    LIST_AMOR = lisr8, [listr8]
                ♦    MASS_GENE = masgen,          [matr_asse_GENE_R]
                ♦    RIGI_GENE = riggen,          [matr_asse_GENE_R]
                ),
   ◊    SANS_CMP =    'LAGR',

                );

if COMB_R and MATR_ASSE:
                [matr_asse_DEPL_R] then [*]  - >  DEPL_R
                [matr_asse_TEMP_R]      [*]  - >  TEMP_R
                [matr_asse_PRES_R]      [*]  - >  PRES_R
                [matr_asse_DEPL_C]      [*]  - >  DEPL_R
                [matr_asse_TEMP_C]      [*]  - >  TEMP_R
                [matr_asse_PRES_C]      [*]  - >  PRES_R
                [matr_asse_GENE_R]      [*]  - >  GENE_R
```

```
if COMB_C and MATR_ASSE:
                  [matr_asse_DEPL_R] then [*]  - >  DEPL_C
                  [matr_asse_TEMP_R]      [*]  - >  TEMP_C
                  [matr_asse_DEPL_C]      [*]  - >  DEPL_C
                  [matr_asse_TEMP_C]      [*]  - >  TEMP_C
                  [matr_asse_PRES_R]      [*]  - >  PRES_C
                  [matr_asse_PRES_C]      [*]  - >  PRES_C

if CALC_AMOR_GENE:
                  [matr_asse_GENE_R] then [*]  - >  GENE_R
```

3 Operands

3.1 Keyword MATR_ASSE

◊ MATR_ASSE

It is possible to crush an object matr_asse with the result of the operation. In this case, it is obligatory to specify here which object is re-used.

3.2 Keyword COMB_R

/ COMB_R

Description of the terms of the linear combination producing a matrix with **real coefficients**.

3.2.1 Operand PART

◊ PART = / 'REAL',
/ 'IMAG',

To carry out extractions or linear combinations of part (S) imaginary (S) or real (S) of complex matrices.

3.2.2 Operand MATR_ASSE

♦ MATR_ASSE = m

Name of the concept matr_asse_* to combine.

3.2.3 Operand COEF_R

♦ COEF_R = R

Real coefficient to apply to the concept argument of MATR_ASSE.

3.3 Keyword COMB_C

/ COMB_C =

Description of the terms of the linear combination producing a matrix with **complex coefficients**.

3.3.1 Recall on the syntax of the complex values

The complex values can be declared in two different ways:

- in the form $a+ib$ with syntax " IH, has, B " where has and B are real numbers,
- in the form $(module, phase)$ with " MP MOD, pH " where MOD and pH are real numbers (pH in degrees).

3.3.2 Operand MATR_ASSE

♦ MATR_ASSE = m

Name of the concept matr_asse_* to combine.

3.3.3 Operands COEF_R/COEF_C

♦ / COEF_R = R

Real coefficient to apply to the concept argument of MATR_ASSE.

/ COEF_C = C

Coefficient complexes to apply to the concept argument of MATR_ASSE.

3.4 Keyword CALC_AMOR_GENE

This keyword makes it possible to build an object of the type matr_asse_gene_R corresponding to the matrix of damping of Basile starting from a list of reduced depreciation, (keyword AMOR_REDUIT or LIST_AMOR).

```
MASS_GENE = masgen, RIGI_GENE = riggen,
```

masgen and riggen are the 2 generalized matrices of mass and rigidity.

3.5 Operand SANS_CMP = 'LAGR'

This operand causes to put in the "zero" terms of the assembled matrix result corresponding to the lines and the columns of the degrees of freedom of Lagrange.

4 Examples of use

4.1 Classical linear combination

```
mat_rs = COMB_MATR_ASSE (COMB_C =   (    _F ( MATR_ASSE = mat_1,
                                           COEF_R = 1.),
                                         _F ( MATR_ASSE= mat_2,
                                           COEF_C= ('IH', 0. , 1. ,),)
                                         ,),)
```

The produced concept mat_rs is of the type matr_asse_*_C (complex):

```
mat_rs = mat_1 + I mat_2
```

4.2 Recopy of a concept of the type matr_asse_*_R

```
mat_sauv = COMB_MATR_ASSE (    COMB_R = _F    (    MATR_ASSE = mat_1,
                                                   COEF_R = 1.))
```

4.3 Difference enters COMB_C and COMB_R:

```
mat_R = COMB_MATR_ASSE (   COMB_R = _F    (    MATR_ASSE = mat_1,
                                                COEF_R = 1.))

#   mat_R      is with real coefficients    mat_R =   mat_1

mat_C = COMB_MATR_ASSE (   COMB_C = _F    (    MATR_ASSE = mat_1,
                                                COEF_R = 1.))

#   mat_C  is with complex coefficients, but the imaginary part is worthless        mat_C =
    mat_1 + I. [0].
```

4.4 Extraction of the real part of a matrix of the type matr_asse_*_C

```
mat_R = COMB_MATR_ASSE     (    COMB_R = _F ( PART = 'REAL',
```

```
                                MATR_ASSE = mat_C,
                                COEF_R = 1. ,),
                            )
```