

Operator ASSE_MATRICE

1 Goal

To create a matrix by assembly of elementary matrices.

The produced matrix is “hollow”; it is stored in form “Morse”.

Product a structure of data of the type `matr_asse_*`.

2 Syntax

```
my [matr_asse_*] = ASSE_MATRICE

( ♦ MATR_ELEM = mel,                                     / [matr_elem_DEPL_R]
                                     / [matr_elem_DEPL_C]
                                     / [matr_elem_TEMP_R]
                                     / [matr_elem_PRES_C]

  ♦ NUME_DDL = naked,                                   [nume_ddl]

  ♦ CHAR_CINE = cha,                                    / [char_cine_meca]
                                     / [char_cine_ther]
                                     / [char_cine_acou]

  ♦ SYME      = 'YES',

  ♦ INFORMATION = / 1,                                [DEFECT]
                  / 2,

)

if MATR_ELEM [matr_elem_DEPL_R] then [*] → DEPL_R
              [matr_elem_DEPL_C]          DEPL_C
              [matr_elem_TEMP_R]          TEMP_R
              [matr_elem_PRES_C]          PRES_C
```

3 Operands

3.1 Operand **MATR_ELEM**

- ◆ `MATR_ELEM = mel,`
Name of the concept `matr_elem_*` to assemble.

3.2 Classification and storage

- ◆ `NUME_DDL = naked,`
Specify the classification of the equations of the assembled system and the storage of the matrix.

3.3 Operand **CHAR_CINE**

- ◇ `CHAR_CINE = cha,`
Name of the kinematic load to take into account in the matrix assembled for a treatment by elimination of the degrees of freedom forced (see order `AFFE_CHAR_CINE` [U4.44.03]).

3.4 Operand **SYME**

with `SYME = 'YES'`

This argument can take only the value `'YES'`. In this case, one forces the symmetrization of the matrix after assembly. Thus, if the matrix created by assembly **K** is not-symmetrical, the keyword `SYME = 'YES'` allows to symmetrize it and replace it by:

$$ma = \frac{1}{2}(K + K^T)$$

3.5 Operand **INFORMATION**

- ◇ `INFORMATION`
The impression of information allows on the assembled matrix

- 1 : pas d' impression,
- 2 : impression amongst stored terms and of the coefficient of conditioning of the degrees of freedom of the type `'LAGR'`,

4 Example

```
subdued = ASSE_MATRICE ( NUME_DDL = naked,  
                          MATR_ELEM = mel,  
                        )
```

Note:

Elementary matrices of mel will be assembled according to classification naked.

*It is necessary thus that this classification takes into account **all** degrees of freedom of these elementary matrices (in particular degrees of freedom corresponding to the dualisation of the boundary conditions). One will thus make:*

```
mel = CALC_MATR_ELEM (OPTION = 'RIGI_MECA',  
                      MODEL = Mo, LOAD = condlim,)  
naked = NUME_DDL (MATR_RIGI = mel)  
subdued = ASSE_MATRICE (NUME_DDL = naked, MATR_ELEM = mel,)
```