

---

## Macro-order ASSEMBLY

---

### 1 Goal

---

To calculate one or more assembled matrices (rigidity, mass, damping...) like one or more assembled vectors.

This order is recommandée to create the concepts néCESSsurfaces with dynamic calculations, which it is the matrices or the vectors. It avoids tiresome calls to the "basic" orders: `CALC_MATR_ELEM [U4.41.01]`, `NUME_DDL [U4.61.01]`, `ASSE_MATRICE [U4.61.22]`, `CALC_VECT_ELEM [U4.61.02]`, `ASSE_VECTEUR [U4.61.23]`.

Product one or more structures of data of the type `matr_asse_*` and `cham_no` more, possibly, a structure of data of the type `nume_ddl`.

## 2 Syntax

```
ASSEMBLY (
  ◆ MODEL = Mo , [model]
  ◆ CHAM_MATER = chmater, [cham_mater]
  ◆ CARA_ELEM = carac , [cara_elem]
  ◆ LOAD = l_char , / [l_char_meca]
  / [l_char_ther]
  / [l_char_acou]
  ◆ INST = / tps , [R]
  / 0. , [DEFECT]
  ◆ CHAR_CINE = l_chci , [l_char_cine]
  ◆ NUME_DDL = naked , [nume_ddl]
  ◆ SOLVEUR = _F (... to see [U4.50.01])

  ◆ # assembly of the matrices:
  / MATR_ASSE = (_F (
    ◆ MATRIX = subdued , [matr_asse_*]
    ◆ OPTION =
      # mechanical phenomenon:
      / 'RIGI_MECA' ,
      ◆ MODE_FOURIER = / nh , [I]
      / 0 , [DEFECT]
      / 'RIGI_GEOM' ,
      ◆ SIEF_ELGA = sig , [cham_elem_SIEF_R]
      ◆ MODE_FOURIER = / nh , [I]
      / 0 , [DEFECT]
      / 'RIGI_ROTA' ,
      / 'IMPE_MECA' ,
      / 'ONDE_FLUI' ,
      / 'RIGI_MECA_HYST' ,
      / 'RIGI_FLUI_STRU' ,
      / 'AMOR_MECA' ,
      / 'MASS_FLUI_STRU' ,
      / 'MECA_GYRO' ,
      / 'RIGI_GYRO' ,
      / 'MASS_MECA' ,
      / 'MASS_MECA_DIAG' ,

      # thermal phenomenon:
      / 'RIGI_THER' ,
      ◆ MODE_FOURIER = / nh , [I]
      / 0 , [DEFECT]
      / 'RIGI_THER_CONV' ,
      / 'RIGI_THER_CONV_D' ,

      # acoustic phenomenon:
      / 'RIGI_ACOU' ,
      / 'MASS_ACOU' ,
      / 'AMOR_ACOU' ,
  ) , ) ,

  # assembly of the vectors:

  / VECT_ASSE = (_F (
```

```
♦ VECTOR = vectas , [cham_no]
♦ OPTION =
# mechanical phenomenon:
  / 'CHAR_MECA' ,
  ♦ LOAD = l_char_v, / [l_char_meca]
  ♦ MODE_FOURIER = / nh , [I]
  / 0 , [DEFECT]
# thermal phenomenon:
  / 'CHAR_THER' ,
  ♦ LOAD = l_char_v, / [l_char_ther]
# acoustic phenomenon:
  / 'CHAR_ACOU' ,
  ♦ LOAD = l_char_v, / [l_char_acou]
),),
♦ TITLE = titr , [l_Kn]
♦ INFORMATION = /1 [DEFECT]
/2
)
```

## 3 Operands

### 3.1 Concepts created by the macro-order

This macro-order creates several typified concepts:

- one `matr_asse_*` with each occurrence of the keyword factor `MATR_ASSE` (keyword `MATRIX`),
- one `cham_no` with each occurrence of the keyword factor `VECT_ASSE` (keyword `VECTOR`),
- possibly one `nume_ddl` (keyword `NUME_DDL`) if this one does not exist first.

That is to say for example the sequence:

...

# `naked` is not a still existing concept.

```
ASSEMBLY (MODEL = Mo, ...
          NUME_DDL = CO ( " naked " ),
          MATR_ASSE = ( _F (MATRICE= CO ("K"), OPTION=' RIGI_MECA'),
          VECT_ASSE = ( _F (VECTEUR= CO ("F"), OPTION=' CHAR_MECA' ))

ASSEMBLY (MODEL = Mo, ...
          NUME_DDL = naked,
          MATR_ASSE = ( _F (MATRICE=CO ( "M" ) , OPTION=' MASS_MECA'),
                      _F (MATRICE=CO ( "With" ) , OPTION=' AMOR_MECA'),
          ), )
```

the 1 <sup>er</sup> call to ASSEMBLY create	1 MATR_ASSE: K 1 VECT_ASSE: F 1 NUME_DDL: naked
2 <sup>ème</sup> call to ASSEMBLY create	2 MATR_ASSE: M and With

#### Note:

*At the time from the 2<sup>ème</sup> call, the concept `nume_ddl` (it is provided exists, it was created by the 1<sup>er</sup> call). In this case, it is supposed to be appropriate for the 2 `matr_asse` to create; it is thus important that it `nume_ddl` that is to say initially calculated for an option of the type " RIGIDITY "to contain the nodes of Lagrange of the loads of blockings (cf. `CALC_MATR_ELEM [U4.61.01]`).*

*In the preceding example, the 3 `matr_asse` `K`, `M` and `With` and the vector `F` are based on the same classification of the unknown factors ( `naked` ). It is a requirement for the algorithms of search for clean modes.*

### 3.2 Operands MODEL / CHAM\_MATER / CARA\_ELEM / INST

- ◆ `MODEL = Mo`

This operand is used to indicate the elements for which must be carried out elementary calculations: it is pointed out that the finite elements for the majority are defined in the model.

There are some exceptions:

- 1) Elements of dualisation of the conditions of DIRICHLET, i.e. elements allowing to impose conditions on the degrees of freedom of displacement in mechanics, degrees of freedom of temperature in thermics and degrees of freedom of pressure in acoustics.
- 2) Nodal heat exchange, loading elements between walls,...

These elements are defined in the concepts of the type `char_meca`, `char_ther` or `char_acou`.

One must thus provide the argument `l_char` for the calculation of the assembled matrices of rigidity: `RIGI_MECA`, `RIGI_THER`, `RIGI_ACOU`, `RIGI_MECA_HYST` and `RIGI_THER_CONV ( _D )`.

*Warning : The translation process used on this website is a "Machine Translation". It may be imprecise and inaccurate in whole or in part and is provided as a convenience.*

Copyright 2017 EDF R&D - Licensed under the terms of the GNU FDL (<http://www.gnu.org/copyleft/fdl.html>)

◇ CHAM\_MATER = chmater

Name of the material field where the characteristics of materials of the elements are defined.

This argument is almost always necessary.

In practice, one can do some:

- for the discrete elements whose elementary matrices are defined in the concept `cara_elem`. See `AFPE_CARA_ELEM` [U4.42.01],
- for the calculation of rigidities due to the dualisation of the boundary conditions.

◇ CARA\_ELEM = carac

Elementary characteristics `carac` are necessary if there exists in the model of the elements of structure (beam, hull or elements discrete) or of the nonisotropic elements of continuous medium.

◇ INST = tps

The argument `tps` is used when the material characteristics or the loadings depend on time. A rather frequent case is that of a mechanical material depend on the temperature which it even depends on time.

### 3.3 Operands LOAD

◇ LOAD = l\_char

This operand has several distinct functions:

- 1) to specify the elements for which the elementary calculations of rigidity (conditions of DIRICHLET) are done,
- 2) for the options `'RIGI_THER_CONV (_D)'` : to give the value the speed of convection,
- 3) for the option `'IMPE_MECA'` : to give the value of the acoustic impedance of the meshes of the edge,
- 4) for the option `'ONDE_FLUI'` : to give the value of the pressure of the incidental wave,
- 5) for the option `'RIGI_ROTA'` : to give the value of the rotation imposed on the model.

#### Note:

*At the time of a call to the macro-order for the assembly of the vectors if the naked concept is entering (under the keyword `NUME_DDL`), the value of the keyword `LOAD l_char` must be identical to that used for the creation of `naked`.*

*That is to say for example the following sequence:*

...

# `naked` is not a still existing concept.

```
ASSEMBLY (MODEL = Mo, LOAD = (char_1, char_2),...
          NUME_DDL = CO ( " naked " ),
          MATR_ASSE = ( _F (MATRICE= CO ("K"), OPTION=' RIGI_MECA'))
```

# `naked` exists and is an entering concept.

```
ASSEMBLY (MODEL = Mo, LOAD = (char_1, char_2),...
          NUME_DDL = naked,
          VECT_ASSE = ( _F (VECTEUR=CO ( "F" ) , OPTION=' CHAR_MECA',
                           CHARGE=CHAR_F),),)
```

### 3.4 Operand CHAR\_CINE

◇ CHAR\_CINE = lchci

This operand allows not “dualiser” certain boundary conditions of Dirichlet (blockings). Those will be eliminated.

## 3.5 Operand **NUME\_DDL** and keyword **SOLVEUR**

◆ `NUME_DDL = naked`

The user the macro one must always give a name of concept behind the keyword `NUME_DDL`. If the concept does not exist, it is created and one uses then `SOLVEUR` (see Ci - afterwards).

If the concept exists, it is used to number the matrices. In this case `SOLVEUR` is ignored.

◇ `SOLVEUR = _F (...)`

This keyword makes it possible to choose the method of resolution of system linear which one will apply to the assembled matrices that one creates in this order. That can seem astonishing to choose as of the construction of the matrices, the way in which possibly one will make use of it! The reason is that the resolution of the linear systems is a “capital” operation, potentially expensive in time and memory capacity (or disc). According to the adopted method, the mode of storage and the method of classification of the unknown factors are given. Those must thus be selected as of the construction of the matrices.

The description of the arguments of this keyword is made in [U4.50.01].

## 3.6 Keyword **MATR\_ASSE**

This keyword factor makes it possible to ask for the calculation of an assembled matrix (with each occurrence).

The name of the assembled matrix is given by the operand `MATRIX` and the “nature” of the matrix is defined by the operand `OPTION`.

Possible options: `'RIGI_MECA'`, `'MASS_MECA'`, ... are described in [U4.61.01].

The choice of the option determines the type of `matr_asse_*` result [U4.61.01].

### 3.6.1 Operands particular to certain options

#### 3.6.1.1 Operand **MODE\_FOURIER**

◇ `MODE_FOURIER = nh`

Positive or null entirety indicating the harmonic of `FOURIER` on whom one calculates the matrices.

By default: `nh = 0`

#### 3.6.1.2 Operand **SIEF\_ELGA** (option `'RIGI_GEOM'`)

◆ `SIEF_ELGA = sig`

The stress field `sig` given for the calculation of the option `'RIGI_GEOM'` must be calculated with the option `'SIEF_ELGA'` (stress field at the points of `GAUSS` of the elements) (cf orders `CALC_CHAM_ELEM` [U4.81.03] or `CALC_CHAMP` [U4.81.04]).

## 3.7 Keyword **VECT\_ASSE**

This keyword factor makes it possible to ask for the calculation of an assembled vector (with each occurrence).

The name of the vector assembled is given by the operand `VECTOR`.

In the possible options of elementary calculation and assembly of vectors, only certain possibilities among all those proposed in the orders `CALC_VECT_ELEM` and `ASSE_VECTEUR` are proposed here: '`CHAR_MECA`' for the mechanical phenomena, '`CHAR_THER`' and '`CHAR_ACOU`' for the thermal and acoustic phenomena respectively.

The whole of options and associated operands are described in [u4.61.02] and [u4.61.23].

**Note:**

*The option 'FORC\_NODA' and the taking into account of a model containing of the substructures under option 'CHAR\_MECA', often used in the case of statics, are not available in this macro-order.*

## 3.7.1 Operands particular to certain options

### 3.7.1.1 Operand `MODE_FOURIER`

◇ `MODE_FOURIER = nh`

Positive or null entirety indicating the harmonic of `FOURIER` on whom one calculates the vectors in the case of the mechanical phenomena.

By default: `nh = 0`

### 3.7.1.2 Operand `LOAD`

◇ `LOAD = l_char_v`

This operand gives access information of loading specific to each vector. Values of `l_char_v` (by addition) the loads supplement `l_char` specified under the keyword `LOAD` apart from the keyword `VECT_ASSE`.

**Note:**

*Doubled blooms in the loads given in `l_char` and in `l_char v` is prohibited.*

Example for the treatment of the loadings:

```
ASSEMBLY (MODEL = Mo,...
          LOAD = (tank_1, char_2) ,
          NUME_DDL = CO ( " naked " ),
          MATR_ASSE = (_F (MATRICE= CO ("K"), OPTION=' RIGI_MECA'),
          VECT_ASSE = (_F (VECTEUR= CO ("F1"),
                          OPTION=' CHAR_MECA', CHARGE=char_A),
                      _F (VECTEUR= CO ("F2"),
                          OPTION=' CHAR_MECA', CHARGE=char_B),
                      _F (VECTEUR= CO ("F3"), OPTION=' CHAR_MECA'),)
```

In the example above, the loads assembled for each vector are:

Vector	Assembled loads
F1	char_1, char_2, char_A
F2	char_1, char_2, char_B
F3	char_1, char_2

## 3.8 Operand `TITLE`

◇ TITLE

Title which one wants to give to the result [U4.03.01].

## 4 Examples

### 4.1 Calculation of mechanical clean modes

```
ASSEMBLY ( MODEL = Mo , LOAD = ch_blocage,
           CHAM_MATER = chmat , CARA_ELEM = carac,
           NUME_DDL = CO( "naked" ),
           MATR_ASSE = ( _F (MATRIX = CO ( "K" ) , OPTION = 'RIGI_MECA' ),
                        _F (MATRIX = CO ( "M" ) , OPTION = 'MASS_MECA' ) , ) ,
           modes = CALC_MODES (MATR_RIGI = K,
                               MATR_MASS = M,
                               ...) )
```

### 4.2 Calculation of modes of buckling of Euler

```
# chsief is the stress field related to the loading

ASSEMBLY ( MODEL = Mo , LOAD = ch_blocage,
           CHAM_MATER = chmat , CARA_ELEM = carac,
           NUME_DDL = CO( "naked" ) ,
           MATR_ASSE = ( _F (MATRICE= CO ("K"), OPTION=' RIGI_MECA'),
                        _F (MATRICE= CO ( "KG" ), OPTION=' RIGI_GEOM',
                            SIEF_ELGA=chsief) , ) ,
           modflamb = CALC_MODES (MATR_RIGI = K,
                                   MATR_MASS = KG,
                                   ...) )
```

### 4.3 Calculation of answer forced

```
ASSEMBLY ( MODEL = Mo , LOAD = ch_blocage,
           CHAM_MATER = chmat , CARA_ELEM = carac,
           NUME_DDL = CO( "naked" ),
           MATR_ASSE = ( _F (MATRIX = CO ( "K" ) , OPTION = 'RIGI_MECA' ),
                        _F (MATRIX = CO ( "M" ) , OPTION = 'MASS_MECA' ) , ) ,
           VECT_ASSE = ( _F (VECTEUR= CO ("F" ),
                             OPTION=' CHAR_MECA', CHARGE=char_A) , )
           answer = DYNA_VIBRA (MATR_RIGI = K, MATR_MASS = M,
                                EXCIT = _F (VECT_ASSE = F, ...) ) )
```