

---

## Macro-command ASSEMBLY

---

### 1 Drank

---

Compute one or more matrixes assembled (stiffness, mass, damping...) like one or more assembled vectors.

This command is recommended to create the concepts necessary to dynamic computations, which it is the matrixes or the vectors. It avoids tiresome calls to the "basic" commands: `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 data structures of the `matr_asse_*` type and `cham_no` more, possibly, a data structure of the `nume_ddl` type.

## 2 Syntax

```

ASSEMBLY (
  ◆MODELE=mo , [model]
  ◇CHAM_MATER=chmater , [cham_mater]
  ◇CARA_ELEM=carac , [cara_elem]
  ◇CHARGE=l_char , / [l_char_meca]
  / [l_char_ther]
  / [l_char_acou]
  [R]
  ◇INST=/tps /0 . , , [DEFAULT]
  ◇CHAR_CINE=l_chci , [l_char_cine]
  ◆NUMÉRIQUE_DDL=nu , [nume_ddl]
  ◇SOLVEUR=_F (... to see [U4.50.01])

  ◆# assembly of the matrixes:
  /MATR_ASSE = (_F (
    ◆MATRICE=matas , [matr_asse_*]
    mechanical
    ◆OPTION= # phenomenon:
    / "RIGI_MECA",
    ◇MODE_FOURIER =/nh , [I]
    /0 , [DEFAULT]
    / "RIGI_GEOM",
    ◆SIEF_ELGA =sig , [cham_elem_SIEF_R]
    ◇MODE_FOURIER =/nh , [I]
    /0 , [DEFAULT]
    / "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",

    # phenomenon thermal:
    / "RIGI_THER",
    ◇MODE_FOURIER =/nh , [I]
    /0 , [DEFAULT]
    / "MASS_THER",
    / "RIGI_THER_CONV",
    / "RIGI_THER_CONV_D",

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

  # assembly of the vectors:
  /VECT_ASSE = (_F (

```

```
♦VECTEUR=vectas , [cham_no]
mechanical
♦OPTION= # phenomenon:
          / "CHAR_MECA",
          ◇CHARGE=l_char_v , / [l_char_meca]
          ◇MODE_FOURIER =/nh , [I]
          /0 , [DEFAULT]
# phenomenon thermal:
          / "CHAR_THER",
          ◇CHARGE=l_char_v , / [l_char_ther]
# phenomenon acoustic:
          / "CHAR_ACOU",
          ◇CHARGE=l_char_v , / [l_char_acou]
          ),),
◇TITER=titr , [l_Kn]
◇INFO=/1 [DEFAULT]
          /2
)
```

## 3 Operands

### 3.1 Concepts created by the macro-command

This macro-command creates several typified concepts:

- a `matr_asse_*` for each occurrence of the key word factor `MATR_ASSE` (key word `MATRICE`),
- a `cham_no` for each occurrence of the key word factor `VECT_ASSE` (key word `VECTEUR`),
- possibly a `nume_ddl` (key word `NUME_DDL`) if this one does not exist as a preliminary.

That is to say for example the sequence:

...

# `nu` is not a still existing concept.

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

```
ASSEMBLY (= Mo MODELS, ...
          NUME_DDL = nu,
          MATR_ASSE = ( _F (MATRICE=CO ( "M" ) , OPTION=' MASS_MECA'),
                      _F (MATRICE=CO ( "A" ) , OPTION=' AMOR_MECA'),
          ),)
```

the 1st call to `ASSEMBLY` creates

```
1 MATR_ASSE: K1
  VECT_ASSE: F1
  NUME_DDL: nu
```

the 2nd call to `ASSEMBLY` creates

```
2 MATR_ASSE: M and A
```

#### Note::

*During the 2nd call, the concept `nume_ddl` is provided (there exists, there was created by the 1st call). In this case, it is supposed to be appropriate for the 2 `matr_asse` to create; it is thus important that the `nume_ddl` is initially calculated for an option of the type "STIFFNESS" 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 `A` and the vector `F` lean on the same classification of the unknowns (`nu`). It is a requirement for the algorithms of search for eigen modes.*

### 3.2 Operands `MODELS` / `CHAM_MATER` / `CARA_ELEM` / `INST`

◆ `MODELE` = `Mo`

This operand is used to indicate the elements for which must be carried out elementary computations: 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 thermal and the degrees of freedom of pressure in acoustics.
- 2) The nodal loading elements, of heat exchange between walls,...

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

One must thus provide the argument `l_char` for the computation of the assembled matrixes of stiffness: `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.*

◇CHAM\_MATER = chmater

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

This argument is almost always necessary.

In practice, one can do some:

- for the discrete elements whose elementary matrixes are defined in the concept `cara_elem`. See `AFFE_CARA_ELEM` [U4.42.01],
- for the computation of the stiffness due to the dualisation of the boundary conditions.

◇CARA\_ELEM = carac

the elementary characteristics `carac` are necessary if there exists in the model structural elements (beam, shell or discrete elements) or elements of continuous medium nonisotropic.

◇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 CHARGE

◇CHARGE = l\_char

This operand has several distinct functions:

- 1) to specify the elements for which elementary computations of stiffness (conditions of DIRICHLET) are done,
- 2) for options "RIGI\_THER\_CONV (\_D)" : to give the value velocity of convection,
- 3) for the option "IMPE\_MECA" : to give the value of the acoustic impedance of meshes of edge,
- 4) for the option "ONDE\_FLUI" : to give the value of the pressure of the incident wave,
- 5) for the option "RIGI\_ROTA" : to give the value of the rotation imposed on the model.

### Note:

*During a call to the macro-command for the assembly of the vectors if the concept `nu` is entering (under key word `NUME_DDL`), the value of key word `CHARGE l_char` must be identical to that used for the creation of `nu`.*

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

...

# `nu` is not a still existing concept.

```
ASSEMBLY (MODELS = Mo, CHARGE = (char_1, char_2),...
          NUME_DDL = CO ( " nu " ),
          MATR_ASSE = ( _F (MATRICE= CO ("K"), OPTION=' RIGI_MECA'))
```

# `nu` exists and is an entering concept.

```
ASSEMBLY (MODELS = Mo, CHARGE = (char_1, char_2),...
          NUME_DDL = nu,
          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 key word solver

◆NUMÉRIQUE\_DDL=nu

the user the macro one must always give a name of concept behind key word NUME\_DDL. If the concept does not exist, it is created and solver then is used (see Ci - afterwards).

If the concept exists, it is used to number the matrixes. In this case solver is ignored.

◇SOLVEUR= \_F (...)

This key word makes it possible to choose the method of resolution of system linear which one will apply to the assembled matrixes that one creates in this command. That can seem astonishing to choose as of the construction of the matrixes, 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 unknowns are given. Those must thus be selected as of the construction of the matrixes.

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

## 3.6 Key word MATR\_ASSE

This key word factor makes it possible to ask for the computation of an assembled matrix (for each occurrence).

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

Possible options: “RIGI\_MECA”, “MASS\_MECA”,... are described in [U4.61.01].

The choice of the option determines the type of the 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 Entier indicating the harmonic of FOURIER on whom one calculates the matrixes.

By default: nh = 0

#### 3.6.1.2 Operand SIEF\_ELGA (option “RIGI\_GEOM”)

◆SIEF\_ELGA = sig

the stress field sig given for the computation of option “RIGI\_GEOM” must be calculated with option “SIEF\_ELGA” (stress field with Gauss points of the elements) (cf commands CALC\_CHAM\_ELEM [U4.81.03] or CALC\_CHAMP [U4.81.04]).

## 3.7 Key word VECT\_ASSE

This key word factor makes it possible to ask for the computation of an assembled vector (for each occurrence).

The name of the vector assembled is given by operand VECTEUR.

In the possible options of elementary computation and assembly of vectors, only certain possibilities among all those proposed in commands `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 set 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 substructures under option "CHAR\_MECA", often used in the case of the static, are not available in this macro-command.*

## 3.7.1 Operands particular to certain options

### 3.7.1.1 Operand `MODE_FOURIER`

◇`MODE_FOURIER` = nh

positive or null Entier 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 `CHARGE`

◇`CHARGE` = l\_char\_v

This operand gives access information of loading specific to each vector. The values of l\_char\_v supplement (by addition) the loads l\_char specified under key word `CHARGE` apart from key word `VECT_ASSE`.

**Note:**

*The duplicates in the loads given in l\_char and l\_char v are prohibited.*

Example for the processing of the loadings:

```
ASSEMBLY (= Mo MODELS, ...
          CHARGE = (tank_1, char_2) ,
          NUME_DDL = CO (" nu " ),
          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	Loads assembled
F1	char_1, char_2, char_A
F2	char_1, char_2, char_B
F3	char_1, char_2

## 3.8 Operand `TITER`

◇TITER

Titrates that one wants to give to result [U4.03.01].

## 4 Examples

### 4.1 Computation of eigen modes mechanics

```
ASSEMBLY ( MODELS = Mo ,          CHARGE = ch_blocage,
           CHAM_MATER = chmat ,    CARA_ELEM = carac,
           NUME_DDL = CO ( "nu" ) ,
           MATR_ASSE = ( _F (MATRICE = CO ( "K" ) , OPTION = "RIGI_MECA" ) ,
                        _F (MATRICE = CO ( "M" ) , OPTION = "MASS_MECA" ) , ) , )

modes = MODE_ITER_SIMULT (MATR_RIGI = K,    MATR_MASS = M, ...)
```

### 4.2 Computation of buckling modes of Eulerian

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

ASSEMBLY ( MODELS = Mo ,          CHARGE = ch_blocage,
           CHAM_MATER = chmat ,    CARA_ELEM = carac,
           NUME_DDL = CO ( "nu" ) ,
           MATR_ASSE = ( _F (MATRICE= CO ( "K" ) , OPTION=' RIGI_MECA' ) ,
                        _F (MATRICE= CO ( "KG" ) , OPTION=' RIGI_GEOM' ,
                            SIEF_ELGA=chsief) , ) , )

modflamb = MODE_ITER_SIMULT (MATR_RIGI = K, MATR_MASS = KG, ...)
```

### 4.3 Computation of forced response

```
ASSEMBLY ( MODELS = Mo ,          CHARGE = ch_blocage,
           CHAM_MATER = chmat ,    CARA_ELEM = carac,
           NUME_DDL = CO ( "nu" ) ,
           MATR_ASSE = ( _F (MATRICE = CO ( "K" ) , OPTION = "RIGI_MECA" ) ,
                        _F (MATRICE = CO ( "M" ) , OPTION = "MASS_MECA" ) , ) ,
           VECT_ASSE = ( _F (VECTEUR= CO ( "F" ) ,
                            OPTION=' CHAR_MECA' , CHARGE=char_A) , )

response = DYNA_VIBRA (MATR_RIGI = K,    MATR_MASS = M,
                      EXCIT = _F (VECT_ASSE = F, ... ) ...)
```