

Manuel d'Utilisation

Fascicule U4.6- : Matrices/Vecteurs élémentaires et assemblage

Document : U4.61.21

Macro commande `MACRO_MATR_ASSE`

1 But

Calculer une ou plusieurs matrices assemblées (rigidité, masse, amortissement, ...).

Cette commande est recommandée pour créer les `matr_asse` nécessaires aux calculs dynamiques. Elle évite de fastidieux appels aux commandes "de base" : `CALC_MATR_ELEM` [U4.41.01], `NUME_DDL` [U4.61.01], `ASSE_MATRICE` [U4.61.22].

Produit une ou plusieurs structure de données de type `matr_asse_*` plus, éventuellement, une structure de données de type `nume_ddl`.

2 Syntaxe

```
MACRO_MATR_ASSE (
  ♦ MODELE      = mo ,                               [modele]
  ◇ CHAM_MATER  = chmater ,                           [cham_mater]
  ◇ CARA_ELEM   = carac ,                             [cara_elem]
  ◇ CHARGE      = l_char ,                             / [l_char_meca]
                                                    / [l_char_ther]
                                                    / [l_char_acou]

  ◇ INST        = / tps ,                             [R]
                  / 0. ,                             [DEFAULT]
  ♦ NUME_DDL     = nu ,                               [nume_ddl]
  ◇ SOLVEUR     = _F (... voir [U4.50.01])

  ♦ MATR_ASSE = (_F (
    ♦ MATRICE    = matas ,                             [matr_asse_*]
    ♦ OPTION     =
      # phénomène mécanique :
      / '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' ,
      / 'RIGI_MECA_LAGR' ,
        ◇ PROPAGATION =  $\alpha$  , [R]
        ♦ THETA      = ch_theta , [theta_geom]
      / 'MASS_MECA' ,
      / 'MASS_MECA_DIAG' ,
      / 'MASS_ID_MDEP_R' ,
      / 'MASS_ID_MDNS_R' ,

      # phénomène thermique :
      / 'RIGI_THER' ,
        ◇ MODE_FOURIER = / nh , [I]
                                      / 0 , [DEFAULT]
      / 'MASS_THER' ,
      / 'RIGI_THER_CONV' ,
      / 'RIGI_THER_CONV_D' ,
      / 'MASS_ID_MTEM_R' ,
      / 'MASS_ID_MTNS_R' ,

      # phénomène acoustique :
      / 'RIGI_ACOU' ,
      / 'MASS_ACOU' ,
      / 'AMOR_ACOU' ,
    ),),
  ◇ TITRE      = titr ,                               [l_Kn]
  )
```

3 Opérandes

3.1 Concepts créés par la macro-commande

Cette macro-commande crée plusieurs concepts typés :

- une `matr_asse_*` à chaque occurrence du mot clé facteur `MATR_ASSE` (mot clé `MATRICE`),
- éventuellement un `nume_ddl` (mot clé `NUME_DDL`) si celui-ci n'existe pas au préalable.

Soit par exemple la séquence :

```
...
# nu n'est pas un concept encore existant.

MACRO_MATR_ASSE (  MODELE = mo ...,
                   NUME_DDL = co("nu"),
                   MATR_ASSE = ( _F( MATRICE= co("K"),
OPTION='RIGI_MECA' ) )

MACRO_MATR_ASSE (  MODELE = mo ...,
                   NUME_DDL = nu,
                   MATR_ASSE = ( _F( MATRICE=co("M"), OPTION='MASS_MECA' ),
                                _F( MATRICE=co("A"), OPTION='AMOR_MECA' ) ) )

le 1er appel à MACRO_MATR_ASSE crée      1 MATR_ASSE : K
                                         1 NUME_DDL : nu
le 2ème appel à MACRO_MATR_ASSE crée      2 MATR_ASSE : M et A
```

Remarque :

Lors du 2^{ème} appel, le concept `nume_ddl` est fourni (il existe, il a été créé par le 1^{er} appel). Dans ce cas, il est supposé convenir aux 2 `matr_asse` à créer ; il est donc important que le `nume_ddl` soit d'abord calculé pour une option de type "RIGIDITE" pour contenir les nœuds de Lagrange des charges de blocages (cf. `CALC_MATR_ELEM` [U4.61.01]).

Dans l'exemple précédent, les 3 `matr_asse` K, M et A s'appuient sur la même numérotation des inconnues (nu). C'est une condition nécessaire pour les algorithmes de recherche de modes propres.

3.2 Opérandes `MODELE` / `CHAM_MATER` / `CARA_ELEM`

♦ `MODELE = mo`

Cet opérande sert à indiquer les éléments pour lesquels doivent être effectués les calculs élémentaires : on rappelle que les éléments finis sont pour la plupart définis dans le modèle.

Il y a quelques exceptions :

- 1) Les éléments de dualisation des conditions de DIRICHLET, c'est-à-dire les éléments permettant d'imposer des conditions sur les degrés de liberté de déplacement en mécanique, les degrés de liberté de température en thermique et les degrés de liberté de pression en acoustique.
- 2) Les éléments de chargement nodal, d'échange thermique entre parois, ...

Ces éléments sont définis dans les concepts de type `char_meca`, `char_ther` ou `char_acou`.

On doit donc fournir l'argument `l_char` pour le calcul des matrices assemblées de rigidité : `RIGI_MECA`, `RIGI_THER`, `RIGI_ACOU`, `RIGI_MECA_HYST`, `RIGI_THER_CONV(_D)` et `RIGI_MECA_LAGR`.

◇ `CHAM_MATER = chmater`

Nom du champ de matériau où sont définies les caractéristiques des matériaux des éléments.

Cet argument est presque toujours nécessaire.

En pratique, on peut s'en passer :

- pour les éléments discrets dont les matrices élémentaires sont définies dans le concept `cara_elem`. Voir `AFFE_CARA_ELEM` [U4.42.01],
- pour le calcul des rigidités dues à la dualisation des conditions aux limites.

◇ `CARA_ELEM = carac`

Les caractéristiques élémentaires `carac` sont nécessaires s'il existe dans le modèle des éléments de structure (poutre, coque ou éléments discrets) ou des éléments de milieu continu non isotropes.

3.3 Opérandes `CHARGE` et `INST`

◇ `CHARGE = lchar`

Cet opérande a plusieurs fonctions distinctes :

- 1) préciser les éléments pour lesquels sont faits les calculs élémentaires de rigidité (conditions de DIRICHLET),
- 2) pour les options de thermo-mécanique, préciser un éventuel champ de température lorsque le matériau est une fonction de la température (cf. mot clé `INST`). Dans ce cas, faire attention à ce qu'il n'y ait pas plusieurs températures différentes dans l'argument `lchar`, la température utilisée est alors la première trouvée dans la liste des charges (cf. `AFFE_CHAR_MECA_...`, opérande `TEMP_CALCULEE` [U4.44.01 §3.3]),
- 3) pour les options `'RIGI_THER_CONV(_D)'` : donner la valeur de la vitesse de convection,
- 4) pour l'option `'IMPE_MECA'` : donner la valeur de l'impédance acoustique des mailles du bord,
- 5) pour l'option `'ONDE_FLUI'` : donner la valeur de la pression de l'onde incidente,
- 6) pour l'option `'RIGI_ROTA'` : donner la valeur de la rotation imposée au modèle.

◇ `INST = tps`

L'argument `tps` n'est utilisé qu'en thermo-mécanique.

Lorsqu'il existe une température dans l'un des concepts `l_char`, on utilise alors le champ de température éventuellement interpolé à l'instant `tps`. Ce champ de température peut faire varier les caractéristiques du champ de matériau si le matériau est une fonction de la température ; sinon cet argument est inutile.

3.4 Opérande `NUME_DDL` et mot clé `SOLVEUR`

◆ `NUME_DDL = nu`

L'utilisateur de la macro doit toujours donner un nom de concept derrière le mot clé `NUME_DDL`. Si le concept n'existe pas, il est créé et on utilise alors `SOLVEUR` (voir ci-après).

Si le concept existe, il est utilisé pour numéroté les matrices. Dans ce cas `SOLVEUR` est ignoré.

◇ `SOLVEUR = _F(...)`

Ce mot clé permet de choisir la méthode de résolution de système linéaire que l'on appliquera aux matrices assemblées que l'on crée dans cette commande. Cela peut sembler étonnant de choisir dès la construction des matrices, la façon dont éventuellement on s'en servira ! La raison en est que la résolution des systèmes linéaires est une opération "capitale", potentiellement coûteuse en temps et en espace mémoire (ou disque). Selon la méthode retenue, le mode de stockage et la méthode de numérotation des inconnues sont déterminés. Ceux-ci doivent donc être choisis dès la construction des matrices.

La description des arguments de ce mot clé est faite dans [U4.50.01].

3.5 Mot clé `MATR_ASSE`

Ce mot clé facteur permet de demander le calcul d'une matrice assemblée (à chaque occurrence).

Le nom de la matrice assemblée est donné par l'opérande `MATRICE` et la "nature" de la matrice est définie par l'opérande `OPTION`.

Les options possibles : '`RIGI_MECA`', '`MASS_MECA`', ... sont décrites dans [U4.61.01 §3.1].

Le choix de l'option détermine le type de la `matr_asse_*` résultat [U4.61.01 §2].

3.6 Opérandes particulières à certaines options

3.6.1 Opérande `MODE_FOURIER`

◇ `MODE_FOURIER = nh`

Entier positif ou nul indiquant l'harmonique de `FOURIER` sur laquelle on calcule les matrices.

Par défaut : `nh = 0`

3.6.2 Opérande `SIEF_ELGA` (option '`RIGI_GEOM`')

◆ `SIEF_ELGA = sig`

Le champ de contraintes `sig` donné pour le calcul de l'option '`RIGI_GEOM`' doit avoir été calculé avec l'option '`SIEF_ELGA_DEPL`' (champ de contraintes aux points de `GAUSS` des éléments) (cf. commandes `CALC_CHAM_ELEM` [U4.81.03] ou `CALC_ELEM` [U4.81.01]).

3.6.3 Opérandes `THETA` et `PROPAGATION` (option '`RIGI_MECA_LAGR`')

Ces opérandes sont utiles à la propagation lagrangienne de fissure.

3.7 Opérande `TITRE`

◇ `TITRE`

Titre que l'on veut donner au résultat [U4.03.01].

4 Exemples

4.1 Calcul de modes propres mécaniques

```
MACRO_MATR_ASSE (  MODELE = 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_A = K      ,  MATR_B = M      ...)
```

4.2 Calcul de modes de flambement d'Euler

% chsief est le champ de contraintes lié au chargement

```
MACRO_MATR_ASSE (  MODELE = 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_A = K      ,  MATR_B = KG      ...)
```

4.3 Calcul de modes propres "numériques"

Les valeurs propres et modes propres calculés ici sont ceux du problème :
 $M - \lambda * I = 0$ où M est une matrice quelconque et I la matrice "identité".

Cette fonctionnalité est une fonctionnalité de "programmeur" qui peut être utile pour comprendre certains comportements numériques : mauvais conditionnement de la matrice de rigidité, ...

```
MACRO_MATR_ASSE (  MODELE = 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("I") , OPTION = 'MASS_ID_MDEP_R'),),)

modflamb = MODE_ITER_SIMULT ( MATR_A = K      ,  MATR_B = I      ...)
```