

**Manuel d'Utilisation****Fascicule U4.6- : Matrices/Vecteurs élémentaires et assemblage****Document : U4.63.11**

---

**Macro commande *MACRO\_PROJ\_BASE***

---

---

**1 But**

---

Projeter des matrices et/ou vecteurs assemblés sur une base modale ou sur une base de RITZ. Les matrices et vecteurs projetés résultats seront utilisés par les algorithmes de calcul en composantes généralisées (*DYNA\_TRAN\_MODAL* [U4.53.21] par exemple).

Cette macro-commande se substitue aux commandes successives suivantes :

- *NUME\_DDL\_GENE* [U4.65.03] qui établit la numérotation des degrés de liberté généralisés,
- une ou plusieurs occurrences de *PROJ\_MATR\_BASE* [U4.63.12] pour projeter une ou plusieurs matrices assemblées,
- une ou plusieurs occurrences de *PROJ\_VECT\_BASE* [U4.63.13] pour projeter un ou plusieurs vecteurs assemblés.

## 2 Syntaxe

```
MACRO_PROJ_BASE
(
  ♦ BASE = ba,                                [mode_meca]
                                              [base_modale]
                                              [mode_gene]

  ♦ NB_VECT = / 9999,                        [DEFAULT]
              / nm,                          [I]

  ♦ MATR_ASSE_GENE = _F( ♦ MATRICE = mt,      [matr_asse_gene_R]
                        ♦ / MATR_ASSE = ma,    [matr_asse_DEPL_R]
                        / MATR_ASSE_GENE = CO('ma'), [matr_asse_gene_R]
                        ♦ PROFIL = / 'DIAG',    [DEFAULT]
                          / 'PLEIN',
                          ),

  ♦ VECT_ASSE_GENE: ( ♦ VECTEUR = vt,          [vect_asse_gene]
                    ♦ TYPE_VECT = / 'FORC',    [DEFAULT]
                      / typ,                  [Kn]
                    ♦ / VECT_ASSE = va,        [cham_no_DEPL_R]
                      / VECT_ASSE_GENE=CO('va'), [vect_asse_gene]
                      ),

  ♦ INFO = / 1,                               [DEFAULT]
            / 2,

)
```

## 3 Opérandes

### 3.1 Opérande `BASE`

- ◆ `BASE = ba`

Concept du type `mode_meca`, `base_modale` ou `mode_gene` (pour la sous-structuration), qui contient les vecteurs définissant le sous-espace de projection.

### 3.2 Opérande `NB_VECT`

- ◇ `NB_VECT = nm`

Nombre de vecteurs utilisés dans la base (on prend les `nm` premiers). On vérifie que le nombre `nm` est bien inférieur au nombre de vecteurs de la base, dans le cas contraire (`nm = 9999`) on utilise tous les vecteurs fournis.

### 3.3 Mot clé `MATR_ASSE_GENE`

- ◇ `MATR_ASSE_GENE`

Mot clé facteur définissant le nom de la matrice projetée résultat et le nom de la matrice à projeter. Ce mot clé doit être répété autant de fois qu'il y a de matrices à projeter.

#### 3.3.1 Opérande `MATRICE`

- ◆ `MATRICE = mt`

Concept du type `matr_asse_gene_R`, matrice généralisée résultat.

#### 3.3.2 Opérandes `MATR_ASSE` / `MATR_ASSE_GENE`

- ◇ / `MATR_ASSE = ma`

Concept du type `matr_asse_DEPL_R`, matrice assemblée que l'on souhaite projeter.

- / `MATR_ASSE_GENE = ma`

Concept du type `matr_asse_gene_R`, matrice assemblée issue de la sous-structuration, que l'on souhaite projeter.

### 3.3.3 Opérande `PROFIL`

◇ `PROFIL = / 'DIAG' [DEFAULT]  
                  / 'PLEIN'`

Cf. `NUME_DDL_GENE` [U4.65.03].

Si une matrice présente un profil 'DIAG' et une autre un profil 'PLEIN', deux numérotations seront créées avec `NUME_DDL_GENE`.

### 3.4 Mot clé `VECT_ASSE_GENE`

◇ `VECT_ASSE_GENE`

Mot clé facteur définissant le nom du vecteur projeté résultat et le nom du vecteur à projeter. Ce mot clé doit être répété autant de fois qu'il y a de vecteurs à projeter.

#### 3.4.1 Opérande `VECTEUR`

◆ `VECTEUR = vt`

Concept du type `vect_asse_gene`, vecteur généralisé résultat.

#### 3.4.2 Opérande `TYPE_VECT`

◇ `TYPE_VECT = typ`

Chaîne de caractères décrivant le type du champ représenté par le vecteur assemblé, par défaut on attend un champ de type force 'FORC', les autres possibilités sont 'DEPL', 'VITE' et 'ACCE'.

#### 3.4.3 Opérandes `VECT_ASSE` / `VECT_ASSE_GENE`

◇ `/ VECT_ASSE = va`

Concept du type `cham_no_DEPL_R`, vecteur assemblé que l'on souhaite projeter.

`/ VECT_ASSE_GENE = va`

Concept du type `vect_asse_gene`, vecteur assemblé issu de la sous-structuration, que l'on souhaite projeter.

### 3.5 Opérande `INFO`

◇ `INFO = / 1 [DEFAULT]  
          / 2`

Niveau d'impression d'informations pour la commande `NUME_DDL_GENE` (cf. [U4.65.03]).

## 4 Phase d'exécution

Les vérifications prévues dans les commandes `NUME_DDL_GENE` [U4.65.03], `PROJ_MATR_BASE` [U4.63.12] et `PROJ_VECT_BASE` [U4.63.13] sont réalisées.

## 5 Exemple d'utilisation

```
# DYNAMIQUE TRANSITOIRE SUR BASE MODALE SYSTEME MASSE ET RESSORT

DEBUT( )

RESSORT=LIRE_MALLAGE( )

MODELE=AFFE_MODELE(   MAILLAGE=RESSORT,
                      AFFE=_F(   TOUT = 'OUI',   PHENOMENE = 'MECANIQUE',
                                MODELISATION = 'DIS_T'   ) )

CHAMPCAR=AFFE_CARA_ELEM(   MODELE=MODELE,
                           DISCRET=(
                               _F(   MAILLE = ('E1','E2','E3','E4','E5','E6','E7','E8','E9'),
                                     CARA = 'K_T_D_L', VALE = (1.E5, 1., 1.)),

                               _F(   MAILLE = 'E1',
                                     CARA = 'A_T_D_L', VALE = (250., 1., 1.)),

                               _F(   MAILLE = ('E2','E3','E4','E5','E6','E7','E8'),
                                     CARA = 'A_T_D_L', VALE = (50., 1., 1.)),

                               _F(   MAILLE = 'E9',
                                     CARA = 'A_T_D_L', VALE = (25., 1., 1.)),

                               _F(   MAILLE=('P1','P2','P3','P4','P5','P6','P7','P8'),
                                     CARA = 'M_T_D_N', VALE = 10.)) )

CHARGE=AFFE_CHAR_MECA(   MODELE=MODELE,DDL_IMPO=(
                           F(   TOUT = 'OUI', DY = 0., DZ = 0.),
                           _F(   NOEUD = ('N1','N10'), DX = 0.0)) )

EFFORT=AFFE_CHAR_MECA(   MODELE=MODELE,
                           FORCE_NODALE=_F(   NOEUD = 'N5',   FX = 1. ) )

RIGIELEM=CALC_MATR_ELEM(   MODELE=MODELE,   OPTION='RIGI_MECA',
                           CARA_ELEM=CHAMPCAR,   CHARGE=CHARGE )

MASSELEM=CALC_MATR_ELEM(   MODELE=MODELE,   OPTION='MASS_MECA',
                           CARA_ELEM=CHAMPCAR,   CHARGE=CHARGE )

AMORELEM=CALC_MATR_ELEM(   MODELE=MODELE,   OPTION='AMOR_MECA',
                           CARA_ELEM=CHAMPCAR,   CHARGE=CHARGE )

EFFOELEM=CALC_VECT_ELEM(   CHARGE=EFFORT,   OPTION='CHAR_MECA',
                           CARA_ELEM=CHAMPCAR )

NUMEROTA=NUME_DDL(   MATR_RIGI=RIGIELEM )

MATRRIGI=ASSE_MATRICE(   MATR_ELEM=RIGIELEM,   NUME_DDL=NUMEROTA )

MATRMAS=ASSE_MATRICE(   MATR_ELEM=MASSELEM,   NUME_DDL=NUMEROTA )

MATRAMOR=ASSE_MATRICE(   MATR_ELEM=AMORELEM,   NUME_DDL=NUMEROTA )

VECTASS=ASSE_VECTEUR(   VECT_ELEM=EFFOELEM,   NUME_DDL=NUMEROTA )

MODES=MODE_ITER_INV(   MATR_A=MATRRIGI,   MATR_B=MATRMAS,
                      CALC_FREQ=_F(   OPTION = 'AJUSTE',
                                      FREQ = ( 5., 10., 15., 20., 24., 27., 30., 32., ) )
                      )
```

Titre : Macro commande MACRO\_PROJ\_BASE  
Auteur(s) : E. BOYERE, Y. PONS, J. PIGAT

Date : 11/02/03  
Clé : U4.63.11-D Page : 6/6

```
FONCTIO1=DEFI_FONCTION(  NOM_PARA='INST',  NOM_RESU='ALPHA',  
                        VALE=( 0., 1., 1., 1., 1.0000001, 0., ),  
                        PROL_DROITE='CONSTANT'  
                        )  
  
MACRO_PROJ_BASE(  BASE=MODES,  
                MATR_ASSE_GENE=(  
                    _F(  
                        MATRICE = CO('MASSEGEN'),  
                        MATR_ASSE = MATRMASS),  
                    _F(  
                        MATRICE = CO('RIGIDGEN'),  
                        MATR_ASSE = MATRRIGI),  
                    _F(  
                        MATRICE = CO('AMORTGEN'),  
                        MATR_ASSE = MATRAMOR,  
                        PROFIL = 'PLEIN'))),  
                VECT_ASSE_GENE=  
                    _F(  
                        VECTEUR = CO('EFFOGENE'),  
                        VECT_ASSE = VECTASS)  
                )  
  
DYNAMODA=DYNA_TRAN_MODAL(  MASS_GENE=MASSEGEN,  RIGI_GENE=RIGIDGEN,  
                           AMOR_GENE=AMORTGEN,  
                           EXCIT=_F(  VECT_GENE = EFFOGENE,  
                                       FONC_MULT = FONCTIO1),  
                           INCREMENT=_F(  INST_INIT = 0.,  
                                           INST_FIN = 1.5, PAS = 0.00001)  
                           )  
  
FIN()
```

# Séquence des commandes équivalentes à la macro-commande MACRO\_PROJ\_BASE :

```
_0000000 = NUME_DDL_GENE    (  BASE=MODES,  
                               STOCKAGE='DIAG'      )  
MASSEGEN = PROJ_MATR_BASE   (  BASE=MODES,  
                               NUME_DDL_GENE=_0000000,  
                               MATR_ASSE=MATRMASS    )  
RIGIDGEN = PROJ_MATR_BASE   (  BASE=MODES,  
                               NUME_DDL_GENE=_0000000,  
                               MATR_ASSE=MATRRIGI    )  
_0000001 = NUME_DDL_GENE    (  BASE=MODES,  
                               STOCKAGE='PLEIN'      )  
AMORTGEN = PROJ_MATR_BASE   (  BASE=MODES,  
                               NUME_DDL_GENE=_0000001,  
                               MATR_ASSE=MATRAMOR    )  
EFFOGENE = PROJ_MATR_BASE   (  BASE=MODES,  
                               NUME_DDL_GENE=_0000001,  
                               VECT_ASSE=VECTASS     )
```