

**Manuel d'Utilisation**  
**Fascicule U4.6- : Matrices/Vecteurs élémentaires et assemblage**  
**Document : U4.61.11**

## Opérateur NUME\_DDL

---

### 1 But

---

Numéroter les inconnues d'un système d'équations linéaires. Cet opérateur "prépare" également l'assemblage de la matrice associée à ce système d'équations.

C'est dans cet opérateur qu'est choisie la méthode de résolution des systèmes linéaires : LDLT, gradient conjugué ou multi-frontale.

Produit une structure de données de type `nume_ddl`.

## 2    Syntaxe

```
nu [nume_ddl] = NUME_DDL
```

```
(  ♦  /  MODELE      = mo      ,      [modele]
      ◇  CHARGE      = lchar    ,      /      [l_char_meca]
                                   /      [l_char_ther]
                                   /      [l_char_acou]

      /  MATR_RIGI =  lmel      ,      /      [l_matr_elem_TEMP_R]
                                   /      [l_matr_elem_DEPL_R]
                                   /      [l_matr_elem_DEPL_C]
                                   /      [l_matr_elem_PRES_C]

      /  METHODE =      'MULT_FRONT' ,      [DEFAULT]

      ◇  RENUM =      /  'METIS' ,      ,      [DEFAULT]
                                   /  'MD' ,
                                   /  'MDA' ,

      /  METHODE =      'LDLT' ,

      ◇  RENUM =      /  'RCMK' ,      [DEFAULT]
                                   /  'SANS' ,

      /  METHODE =      'GCPC' ,

      ◇  RENUM =      /  'RCMK' ,      [DEFAULT]
                                   /  'SANS' ,

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

### 3 Généralités

Cette commande permet de numéroter les équations (et les inconnues) des systèmes linéaires à résoudre. On prépare également le travail d'assemblage des matrices, en établissant les tableaux de pointeurs correspondant au stockage choisi pour ces matrices.

Le choix de la méthode est fait par le mot clé `METHODE` (méthode '`MULT_FRONT`' par défaut).

## 4 Opérandes

### 4.1 Opérande `MODELE` / `CHARGE`

♦ / `MODELE` = `mo` , `CHARGE` = `lchar`

Le code numérotera les degrés de liberté des éléments du modèle `mo` ainsi que les degrés de liberté de dualisation des conditions cinématiques dualisées éventuellement présentes dans les charges de la liste `lchar`.

### 4.2 Opérande `MATR_RIGI`

♦ / `MATR_RIGI` = `lmel`

Liste des `matr_elem_*` permettant d'établir la numérotation des inconnues du problème.

Le but de cet opérateur est de numéroter tous les degrés de liberté du problème. Ces degrés de liberté sont ceux portés par les matrices élémentaires calculées par l'opérateur `CALC_MATR_ELEM(lmel)`.

Du fait de la dualisation des conditions "cinématiques", les degrés de liberté de Lagrange sont portés par les `matr_elem` calculés par l'option '`RIGI_MECA`' (ou '`RIGI_THER`', ...) sur les charges où sont définies les conditions cinématiques.

Il est donc important de donner dans `lmel`, la liste de tous les `matr_elem` de "rigidité" et de ne pas oublier les charges dans l'opérateur `CALC_MATR_ELEM` [U4.61.01].

### 4.3 Opérande `METHODE`

Cet opérande sert à choisir la méthode de résolution que l'on appliquera aux futurs systèmes linéaires qui seront construits sur le `nume_ddl` produit par cette commande.

```
/ 'MULT_FRONT'  méthode "multi_frontale"  
/ 'LDLT'        factorisation de type LU sans pivotage  
/ 'GCPC'        gradient conjugué
```

Le choix de la méthode de résolution a une influence importante sur le mode de stockage des matrices assemblées (futures) `matr_asse_*`

```
/ 'LDLT'        stockage "ligne de ciel" par blocs  
/ 'MULT_FRONT' ou 'GCPC'  stockage "morse"
```

Le stockage morse (qui ne conserve que les termes non nuls de la matrice) est beaucoup plus économique que le stockage ligne de ciel.

En revanche pour un gros système, ce stockage impose une taille **mémoire** importante car la matrice est stockée dans un seul vecteur, alors que le stockage ligne de ciel est découpé en plusieurs blocs de taille fixée par le mot clé `TAILLE_BLOC` de la commande `DEBUT`.

## 4.4 Opérande RENUM

Cet opérande sert à renuméroter (si on le souhaite) les inconnues du problème.

- 'MD'        (Minimum Degré) : c'est la méthode de renumérotation associée à la résolution multi-frontale. Elle minimise le remplissage de la matrice lors de sa factorisation (c'est-à-dire : apparition de termes non nuls).
- 'MDA'       (Minimum Degré Approchée) : c'est une méthode approchée de la méthode précédente. A priori, la numérotation obtenue est un peu moins optimale que par 'MD' (pour ce qui concerne le remplissage), mais elle est moins coûteuse à établir.
- 'METIS'     autre méthode de numérotation pour la méthode multi-frontale. Elle est basée sur une technique de "dissection emboîtée".
- 'SANS'       on garde l'ordre initial des nœuds donné dans le fichier de maillage
- 'RCMK'       "Reverse CUTHIL MAC KEE" cet algorithme de renumérotation est souvent efficace pour réduire la place nécessaire au stockage "ligne de ciel" de la matrice assemblée. Il est donc aussi efficace en temps CPU lors de la triangulation d'une telle matrice, par l'opérateur FACT\_LDLT [U4.55.01].

## 4.5 Opérande INFO

- ◇    INFO =
- /    1    sur le fichier MESSAGE.
  - nombre total de degrés de liberté, nombre de nœuds,
  - nombre de degrés de liberté de LAGRANGE,
  - hauteur maximum d'une colonne (et hauteur moyenne),
  - nombre de termes stockés (pour le stockage 'MORSE'),
  - nombre de blocs (pour le stockage ligne de ciel),
  - informations concernant la renumérotation.

## 5 Exemples

```
# pour une résolution "LDLT" avec renumérotation "RCMK" :
nu = NUME_DDL      (MATR_RIGI = mel, METHODE = 'LDLT' ,
K   = ASSE_MATRICE (MATR_ELEM = mel, NUME_DDL = nu ,
K   = FACT_LDLT    (MATR_ASSE = K,   reuse = K,
U   = RESO_LDLT    (MATR_ASSE = K,   CHAM_NO = F,

# pour une résolution "MULTI_FRONTALE" :
nu = NUME_DDL      (MATR_RIGI = mel,
K   = ASSE_MATRICE (MATR_ELEM = mel, NUME_DDL = nu ,
K   = FACT_LDLT    (MATR_ASSE = K,   reuse = K,
U   = RESO_LDLT    (MATR_ASSE = K,   CHAM_NO = F,

# pour une résolution "GRADIENT CONJUGUE" :
nu = NUME_DDL      (MATR_RIGI = mel, METHODE = 'GCPC' ,
K   = ASSE_MATRICE (MATR_ELEM = mel, NUME_DDL = nu ,
K2  = FACT_GRAD    (MATR_ASSE = K,
U   = RESO_GRAD    (MATR_ASSE = K,   MATR_FACT = K2, CHAM_NO = F,
```