

**Manuel d'Utilisation**  
**Fascicule U4.5- : Méthodes de résolution**  
**Document : U4.55.01**

## Opérateur *FACT\_LDLT*

### 1 But

Factoriser une matrice assemblée en un produit de deux matrices triangulaires. Cette factorisation permet ensuite de résoudre rapidement les systèmes linéaires grâce à la commande *RESO\_LDLT*.

Cette commande n'est utilisable que pour les matrices dont la méthode de résolution (choisie dans la commande *NUME\_DDL*) est '*LDLT*', '*MULT\_FRONT*' ou '*MUMPS*'.

Produit (ou modifie) une structure de données de type *matr\_asse*.

## 2    Syntaxe

```

mat [matr_asse_*] =   FACT_LDLT

(   ◇   reuse = mat ,
      ◆   MATR_ASSE =       mat ,                   /           [matr_asse_DEPL_R]
                                                     /           [matr_asse_DEPL_C]
                                                     /           [matr_asse_TEMP_R]
                                                     /           [matr_asse_TEMP_C]
                                                     /           [matr_asse_PRES_R]
                                                     /           [matr_asse_PRES_C]

/   #   Si méthode LDLT ou MULT_FRONT:
      ◇   STOP_SINGULIER =       /   'OUI' ,           [DEFAULT]
                                     /   'NON' ,

      ◇   NPREC =       /   nprec ,           [I]
                             /   8 ,           [DEFAULT]

      ◇   #   Si méthode LDLT :
          ◇   PRE_COND =       /   'SANS' ,           [DEFAULT]
                                     /   'DIAG' ,

          ◇   /   BLOC_DEBUT = bd ,           [I]
                     /   DDL_DEBUT = dd ,           [I]

          ◇   /   BLOC_FIN =   bf ,           [I]
                     /   DDL_FIN =   df ,           [I]

/   #   Si méthode MUMPS:
      ◇   TYPE_RESOL =       /   'AUTO' ,           [DEFAULT]
                                     /   'NONSYM' ,
                                     /   'SYMDEF' ,
                                     /   'SYMGEM' ,

      ◇   PCENT_PIVOT =       /   20 ,           [DEFAULT]
                                     /   pcpiv           [R]

      ◇   TITRE =   titre ,           [l_K80]

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

)

si MATR_ASSE :           [matr_asse_DEPL_R]   alors   [*]   -> DEPL_R
                          [matr_asse_DEPL_C]           DEPL_C
                          [matr_asse_TEMP_R]           TEMP_R
                          [matr_asse_TEMP_C]           TEMP_C
                          [matr_asse_PRES_R]           PRES_R
                          [matr_asse_PRES_C]           PRES_C

```

### 3 Opérandes

Le choix de la méthode de résolution est fait au préalable dans la commande `NUME_DDL` (mot clé `METHODE`).

Trois méthodes sont possibles : `'LDLT'`, `'MULT_FRONT'` et `'MUMPS'`.

- pour la méthode `'LDLT'`, si l'utilisateur utilise un concept réentrant :  
    `matas = FACT_LDLT (reuse = matas, MATR_ASSE = matas)`  
il "écrase" sa matrice initiale. Il ne pourra donc plus faire de combinaison de matrices ou de produit matrice x vecteur,
- pour les méthodes `'MULT_FRONT'` et `'MUMPS'`, la matrice factorisée est stockée ailleurs. Pour ces méthodes, il est donc fortement recommandé d'utiliser un concept réentrant ; cela évite de dupliquer la matrice initiale.
- pour la méthode `'MUMPS'`, la matrice factorisée n'est stockée qu'en mémoire. Quand le "job" se termine, la factorisée est perdue. Il faut donc re-factoriser les matrices à chaque `POURSUITE`.

#### 3.1 Opérande `MATR_ASSE`

♦ `MATR_ASSE = mat`

Nom de la matrice assemblée à factoriser. Cette matrice peut être réelle ou complexe, symétrique ou non.

#### 3.2 Opérandes `STOP_SINGULIER`, `NPREC`, `TYPE_RESOL` et `PCENT_PIVOT`

Ces mots clés sont décrits dans [U4.50.01].

#### 3.3 Opérande `TITRE`

♦ `TITRE = titre`

Titre que l'on veut donner au résultat [U4.02.01].

#### 3.4 Opérande `INFO`

♦ `INFO =`

1 : pas d'impression

#### 3.5 Opérande `PRE_COND`

L'effet de ce mot clé est de "normaliser" la matrice avant sa factorisation. L'auteur de ce document n'en a jamais vu l'intérêt !

♦ `PRE_COND =`

Type du préconditionnement.

'SANS' : pas de préconditionnement (défaut).

'DIAG' : préconditionnement par la diagonale. Les lignes ayant un terme diagonal nul, ne sont pas traitées.

### 3.6 Factorisation partielle (méthode LDLT)

Pour la méthode 'LDLT', l'opérateur permet de ne factoriser que partiellement la matrice. Cette possibilité est "historique". Elle permet de factoriser la matrice en plusieurs "fois" (plusieurs travaux). Aujourd'hui, on n'imagine pas bien l'intérêt de cette fonctionnalité.

◇ / BLOC\_DEBUT = bd  
bd : factorisation partielle depuis le bd<sup>-ième</sup> bloc inclus.

/ DDL\_DEBUT = dd  
dd : factorisation partielle depuis la dd<sup>-ième</sup> équation incluse (en numérotation interne établie par l'opérateur NUME\_DDL [U4.61.11]).

◇ / BLOC\_FIN = bf  
bf : factorisation partielle jusqu'au bf<sup>-ième</sup> bloc inclus.

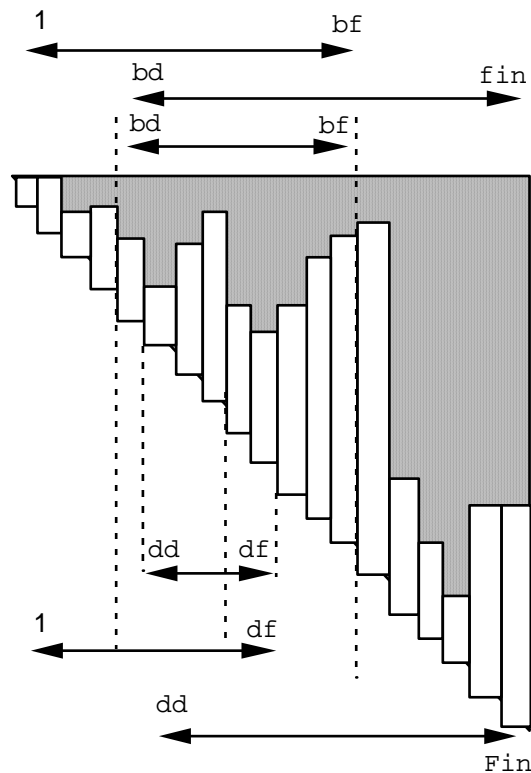
/ DDL\_FIN = df  
df : factorisation partielle jusqu'à la df<sup>-ième</sup> équation incluse (en numérotation interne établie par l'opérateur NUME\_DDL [U4.61.11]).

L'opérateur effectue les vérifications suivantes :

- si le nom du concept produit correspond à un nouveau concept, la factorisation est hors-place : on vérifie que la factorisation débute bien au premier degré de liberté.
- si l'argument df est négatif ou supérieur à l'ordre de la matrice, alors celle-ci sera factorisée jusqu'à la fin. Sinon, on effectue une factorisation partielle (de la 1<sup>ère</sup> équation) jusqu'à la df<sup>-ième</sup> équation incluse.

BLOC\_DEBUT et DDL\_DEBUT

- en l'absence des mots clés BLOC\_DEBUT et DDL\_DEBUT, la matrice sera factorisée à partir de sa première ligne.
- si l'argument bd du mot clé BLOC\_DEBUT est négatif ou nul, la matrice sera factorisée à partir du premier bloc. Sinon, on effectue une factorisation partielle à partir du bd<sup>-ième</sup> bloc inclus.
- si l'argument dd du mot clé DDL\_DEBUT est négatif ou nul, la matrice sera factorisée à partir de la première équation. Sinon, on effectue une factorisation partielle à partir de la dd<sup>-ième</sup> équation incluse.



## 4 Exemple

### 4.1 Méthode LDLT hors place

```
nu = NUME_DDL      (MATR_RIGI = kel,   METHODE = 'LDLT')  
k  = ASSE_MATRICE (MATR_ELEM = kel,   NUME_DDL = nu)  
k2 = FACT_LDLT    (MATR_ASSE = k)  
U  = RESO_LDLT    (MATR_FACT = k2,    CHAM_NO = F)
```

### 4.2 Méthode LDLT en place

```
nu = NUME_DDL      (MATR_RIGI = kel,   METHODE = 'LDLT')  
k  = ASSE_MATRICE (MATR_ELEM = kel,   NUME_DDL = nu)  
k  = FACT_LDLT    (reuse = k, MATR_ASSE = k)  
U  = RESO_LDLT    (MATR_FACT = k,     CHAM_NO = F)
```

### 4.3 Méthode multi-frontale

```
nu = NUME_DDL      (MATR_RIGI = kel,   METHODE = 'MULT_FRONT')  
k  = ASSE_MATRICE (MATR_ELEM = kel,   NUME_DDL = nu)  
k  = FACT_LDLT    (reuse = k, MATR_ASSE = k)  
U  = RESO_LDLT    (MATR_FACT = k,     CHAM_NO = F)
```

Page laissée intentionnellement blanche.