

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

Opérateur **FACTORISER**

1 But

Factoriser une matrice assemblée.

La factorisation produite par cet opérateur est souvent fournie à l'opérateur **RESOUDRE** [U4.55.02] pour résoudre les systèmes linéaires.

Cet opérateur permet :

- soit de factoriser la matrice assemblée en un produit de deux matrices triangulaires,
- soit de construire une matrice de préconditionnement pour une résolution par gradient conjugué dans le but d'accélérer la convergence lors de la résolution.

Cette commande n'est utilisable que pour les matrices dont la méthode de résolution (choisie dans la commande **NUME_DDL**) est 'LDLT', 'MULT_FRONT', 'MUMPS' ou 'GCPC'

Produit (ou modifie) une structure de données de type `matr_asse`.

2 Syntaxe

```

mat [matr_asse_*] =    FACTORISER

(    ◇   reuse = mat ,

     ♦   MATR_ASSE =        mat ,
        #   Si méthode MULT_FRONT, MUMPS, LDLT :

                                        /                [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 GCPC :

                                        /                [matr_asse_DEPL_R]
                                        /                [matr_asse_TEMP_R]
                                        /                [matr_asse_PRES_R]

     /   #   Si méthode MULT_FRONT :
         ◇   STOP_SINGULIER =        /   'OUI'   ,   [DEFAULT]
                                        /   'NON'   ,
         ◇   NPREC =        /   nprec   ,   [I]
                                        /   8       ,   [DEFAULT]

     /   #   Si méthode MUMPS :
         ◇   TYPE_RESOL =        /   'AUTO'   ,   [DEFAULT]
                                        /   'NONSYM'   ,
                                        /   'SYMDEF'   ,
                                        /   'SYMGEM'   ,
         ◇   PCENT_PIVOT =        /   20   ,   [DEFAULT]
                                        /   pcpiv       ,   [R]
         ◇   PRETRAITEMENTS = /   'AUTO'   ,   [DEFAULT]
                                        /   'SANS'   ,   [R]
         ◇   ELIM_LAGR2 =        /   'OUI'   ,   [DEFAULT]
                                        /   'NON'   ,   [R]

     /   #   Si méthode GCPC :
         ◇   PRE_COND =        'LDLT_INC' ,   [DEFAULT]
         ◇   NIVE_REEMPLISSAGE = /   0 ,   [DEFAULT]
                                        /   n ,   [I]

     /   #   Si méthode LDLT :
         ◇   STOP_SINGULIER =        /   'OUI'   ,   [DEFAULT]
                                        /   'NON'   ,
         ◇   NPREC =        /   nprec   ,   [I]
                                        /   8       ,   [DEFAULT]
         ◇   /   BLOC_DEBUT = bd ,   [I]
                /   DDL_DEBUT = dd ,   [I]
         ◇   /   BLOC_FIN = bf ,   [I]
                /   DDL_FIN = df ,   [I]

     ◇   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).

Quatre méthodes sont possibles : 'MULT_FRONT', 'MUMPS', 'GCPC' et 'LDLT',

3.1 Mot clé reuse = matas

- pour les méthodes 'MULT_FRONT', 'MUMPS' et 'LDLT' la matrice factorisée est stockée en plus de la matrice initiale.
Il est fortement recommandé pour ces méthodes d'utiliser un concept réentrant afin d'éviter 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.
- Pour la méthode 'GCPC', il n'est pas permis d'utiliser un concept réentrant.

3.2 Opérande MATR_ASSE

♦ MATR_ASSE = mat

Nom de la matrice assemblée à factoriser ou à préconditionner selon la méthode.

Pour les méthodes 'LDLT', 'MULT_FRONT' et 'MUMPS', cette matrice peut être réelle ou complexe, symétrique ou non. Par contre pour la méthode 'GCPC', cette matrice doit forcément être réelle symétrique.

3.3 Opérandes STOP_SINGULIER, NPREC, TYPE_RESOL, PCENT_PIVOT, PRETRAITEMENTS et ELIM_LAGR2

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

3.4 Opérande TITRE

♦ TITRE = titre

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

3.5 Opérande INFO

♦ INFO =

1 : pas d'impression

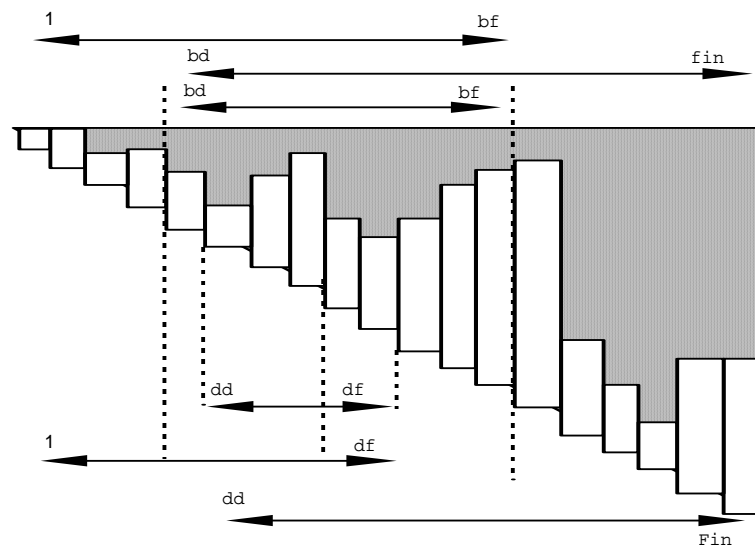
3.6 Factorisation partielle (méthode LDLT)

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^{-ième} bloc inclus.
- / DDL_DEBUT = dd
dd : factorisation partielle depuis la dd^{-ième} équation incluse (en numérotation interne établie par l'opérateur NUME_DDL [U4.61.11]).
- ◇ / BLOC_FIN = bf
bf : factorisation partielle jusqu'au bf^{-ième} bloc inclus.
- / DDL_FIN = df
df : factorisation partielle jusqu'à la df^{-ième} équation incluse (en numérotation interne établie par l'opérateur NUME_DDL [U4.61.11]).

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^{-ième} 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^{-ième} équation incluse.



3.7 Opérande PRE_COND

◇ PRE_COND

Mot-clé spécifique à la méthode 'GCPC'.

Il permet d'affecter la méthode de préconditionnement 'LDLT_INC' :

'LDLT_INC' : la matrice de préconditionnement est obtenue par une décomposition LDLT incomplète de la matrice assemblée.
Celle décomposition est plus ou moins complète, suivant le niveau de remplissage. La matrice résultat `matfac` est de type `matr_asse`.

3.8 Opérande NIVE_REMPLISSAGE

◇ NIVE_REMPLISSAGE = / 0
 / n

Mot-clé spécifique à la méthode 'GCPC'.

La matrice de préconditionnement (P) utilisée pour accélérer la convergence du gradient conjugué est obtenue en factorisant de façon plus ou moins complète la matrice initiale (A).

Si `niv` = 0 (défaut)

P a le même stockage que A. La factorisation est incomplète car on n'utilise pour les calculs que les termes que l'on peut stocker dans P.

P représente donc une approximation (médiocre) de A^{-1} ; son stockage est faible.

Si `niv` = 1

On stocke dans P en plus des termes qui avaient leur place dans le stockage initial, les "descendants" de première génération des termes initiaux. En effet lors de la factorisation, un terme nul dans A peut devenir non nul dans P. On obtient ainsi le remplissage de niveau 1.

Si `niv` = 2, ...

Le même procédé est repris : la matrice P remplie au niveau `niv-1` crée les termes de la matrice P au niveau `niv`.

Plus `niv` est grand, plus la matrice P est proche de A^{-1} et donc plus le gradient conjugué converge vite (en nombre d'itérations).

En revanche, plus `niv` est grand plus le stockage de P devient volumineux (en mémoire et sur disque) et plus les itérations sont coûteuses en CPU.

Les premiers essais ont montré (approximativement) que la taille de P valait :

- 3,5* taille (A) pour `niv` = 1
- 7,5* taille (A) pour `niv` = 2

Notre expérience de ce mot clé est encore limitée et nous conseillons d'utiliser la valeur par défaut (`niv` = 0).

Si `niv` = 0 ne permet pas au gradient conjugué de converger, on essaiera successivement les valeurs `niv` = 1, 2, 3.

4 Exemples

4.1 Méthode multi-frontale

```
nu = NUME_DDL      (MATR_RIGI=kel, METHODE='MULT_FRONT')  
k  = ASSE_MATRICE (MATR_ELEM=kel, NUME_DDL=nu)  
k  = FACTORISER   (reuse=k, MATR_ASSE=k)  
U  = RESOUDRE     (MATR= k , CHAM_NO=F)
```

4.2 Méthode GCPC

```
nu = NUME_DDL      (MATR_RIGI= mel, METHODE= 'GCPC' ) ;  
k  = ASSE_MATRICE (MATR_ELEM= mel, NUME_DDL= nu ) ;  
kp = FACTORISER   (MATR_ASSE= k ) ;  
U  = RESOUDRE     (MATR=k, MATR_PREC = kp , CHAM_NO = F)
```