Opérateur FACTORISER

1 But

Factoriser une matrice assemblée ou **fabriquer une matrice de pré-conditionnement**.

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

Cet opérateur permet:
1. Soit de factoriser la matrice assemblée (en un produit de deux matrices triangulaires) pour les méthodes directes,
2. Soit de construire une matrice de pré conditionnement pour les méthodes itératives dans le but d'accélérer la convergence lors de la résolution.

Cette commande est un préalable obligatoire avant d'appeler la commande de résolution RESOUDRE.

Produit (ou enrichit) une structure de données de type matr_asse.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

2 Syntaxe

```
mat [matr_asse_*] =  FACTORISER

          (  # Obligatoire (sauf si GCPC + LDLT_INC : interdit alors)
             ◊  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  ou PETSC :
                                          /         [matr_asse_DEPL_R]
                                          /         [matr_asse_TEMP_R]
                                          /         [matr_asse_PRES_R]

                 # Si méthode MULT_FRONT :
                   ◊  STOP_SINGULIER   =      /   'OUI', [DEFAUT]
                                              /   'NON'.
                   ◊  RENUM  =                /   'METIS',  [DEFAUT]
                                              /   'MDA',
                                              /   'MD'
                   ◊  NPREC  =  /  nprec  ,           [I]
                                /  8.                 [DEFAUT]

                 # Si méthode MUMPS:  [DEFAUT]
                   ◊  RENUM=          /   'AUTO', [DEFAUT]
                                      /   'AMD',
                                      /   'AMF',
                                      /   'QAMD',
                                      /   'PORD',
                                      /   'METIS',
                                     /    'SCOTCH'
                    ◊  STOP_SINGULIER   =  /   'OUI'  ,   [DEFAUT]
                                           /   'NON'  ,
                    ◊  TYPE_RESOL       =  /   'AUTO' ,   [DEFAUT]
                                           /   'NONSYM',
                                           /   'SYMDEF',
                                           /   'SYMGEM'.
                    ◊  PCENT_PIVOT      =  /   20  ,      [DEFAUT]
                                           /   pcpiv.     [R]
                    ◊  PRETRAITEMENTS   =  /   'AUTO' ,   [DEFAUT]
                                           /   'SANS'.
                    ◊  ELIM_LAGR        =  /   'LAGR2'    ,   [DEFAUT]
                                           /   'NON'.
                    ◊  GESTION_MEMOIRE  =  /   'IN_CORE' ,[DEFAUT]
                                           /   'OUT_OF_CORE',
                                           /   'EVAL'.
#Paramètres limités à certaines versions de MUMPS
                ◊  ACCELERATION     =      /'AUTO',            [DEFAUT]
                                         /  /'FR',
```


Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

```
                                    /   /'FR+',
                                    /   /'LR',
                                    /   /'LR+'

                  ◊ LOW_RANK_SEUIL=      / 0.0                    [DEFAUT]
                                         / lr_seuil   [R]
                # Si méthode GCPC ou PETSC :
                  ◊ / PRE_COND    = /   'LDLT_INC',               [DEFAUT]
                          ◊  NIVE_REMPLISSAGE =  /   0,           [DEFAUT]
                                                 /   niv [ I ]
                          ◊  REMPLISSAGE      =  /   1.,          [DEFAUT]
                                                 /   cr.       [R]
                     /  PRE_COND   = /   'LDLT_SP',
                                     /   'LDLT_DP',
                          ◊  PCENT_PIVOT      =  /   20,          [DEFAUT]
                                                 /   pcent [R]
                          ◊  REAC_PRECOND     =  /   30,          [DEFAUT]
                                                 /   reac, [I]
                          ◊  GESTION_MEMOIRE  =  /   'AUTO',      [DEFAUT]
                                                  /  'IN_CORE',
                          ◊  LOW_RANK_SEUIL   =  /   0.0          [DEFAUT]
                                                 /    lr_seuil [R]
                          ◊   RENUM     =        /    ' SANS '    [DEFAUT]
                                                 /    ' PARMETIS '
                  ◊ / PRE_COND    =   / 'JACOBI',
                                      /  'SOR',
                # Si méthode LDLT :
                        ◊  STOP_SINGULIER  =  /   'OUI'  ,  [DEFAUT]
                                              /  'NON'  ,
                        ◊  NPREC           =  /   nprec  .  [I]
                                              /   8      ,  [DEFAUT]
                        ◊  /   BLOC_DEBUT  =  bd ,         [I]
                           /   DDL_DEBUT   =  dd  ,        [I]
                        ◊  /   BLOC_FIN    =  bf ,         [I]
                           /   DDL_FIN     =  df. ,        [I]

                  ◊  TITRE  =  titre  ,                     [l_K80]

                  ◊  INFO   =  /  1   ,                     [DEFAUT]
                               /  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
```


Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

3 Opérandes

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

Cinq méthodes sont possibles: 'MULT_FRONT', 'MUMPS', 'GCPC', 'PETSC' et 'LDLT'.

3.1 Mot clé reuse = matas

Par souci de simplification de la programmation et aussi pour gagner de l'espace disque, la factorisation « en place » de la matrice (usage de reuse) est en général obligatoire.

L' exception est le couple ('GCPC', 'LDLT_INC'). Dans ce cas, il est interdit d'utiliser reuse.

3.2 Opérande MATR_ASSE

♦ MATR_ASSE = mat

Nom de la matrice assemblée à factoriser ou à pré conditionner selon la méthode.

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

3.3 Opérandes STOP_SINGULIER, NPREC, TYPE_RESOL, PCENT_PIVOT, PRETRAITEMENTS, GESTION_MEMOIRE, ACCELERATION, LOW_RANK_SEUIL, PRE_COND, NIVE_REMPLISSAGE, REMPLISSAGE, RENUM, REAC_PRECOND et ELIM_LAGR

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 permet de factoriser la matrice en plusieurs fois (plusieurs travaux) voire de modifier à la volée les dernières lignes de cette factorisée. Aujourd'hui, cette fonctionnalité présente peu d'intérêt

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

hormis pour certaines méthodes (dite discrètes) de contact-frottement où l'on a, à dessein, placé dans les dernières lignes de la matrice les termes concernant les nœuds susceptibles d'être en contact. Ainsi, au fur et à mesure des itérations d'appariement, les relations entre ces nœuds changeant, on efface puis recalcule que ces dernières contributions de la factorisée. C'est un exemple typique où l'usage astucieux d'un algorithme assez frustre peu amener des gains majeurs (en temps).

◊ / 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

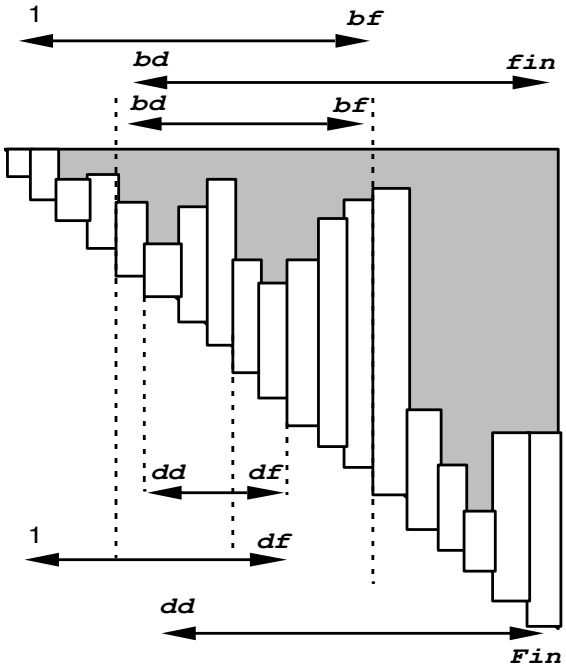
bf : factorisation partielle jusqu'au bf-ième bloc inclus.

/ 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.



4 Exemples

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

Voir les exemples dans la documentation de la commande RESOUDRE [U4.55.02].

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)