Structure de Données sd_matr_asse

Résumé :

Ce document décrit les structures de données sd_matr_asse : les matrices creuses.

Table des matières

1 Généralités

Les objets de type sd_matr_asse représentent les matrices assemblées carrées (au sens des éléments finis). Ce sont en général de gros objets. Ces matrices sont creuses, ce qui explique que leur structure ne soit pas simplement un tableau carré.

Une sd_matr_asse peut provenir d'un assemblage de sd_matr_elem ou d'une combinaison linéaire d'autres sd_matr_asse.

Les tableaux décrivant le stockage de la sd_matr_asse sont dans la structure sd_stockage d'un sd_nume_ddl [D4.06.07].

Il existe des sd_matr_asse symétriques et des sd_matr_asse non-symétriques. Mais on suppose que la topologie de la matrice est toujours symétrique. C'est à dire que les termes non nuls sont disposés symétriquement par rapport à la diagonale.

2 Arborescences

```
sd_matr_asse  (K19)  ::=record

(o)   '.REFA'          :       OJB      S   V  K24

/ # Cas général (solveur /= FETI i.e. REFA(5) /= 'FETI') :
    (o)   '.VALM'    :       OJB      XD  V  R/C   NUM() nbobj=1/2

    (f) # si il existe des ddls de Lagrange :
       (o)   '.CONL' :       OJB      S   V  R

    (f) # si la sd_matr_asse provient de l'assemblage de matrices élémentaires :
       (o)   '.LIME' :       OJB      S   V  K8

    (f) # si la sd_matr_asse a été "factorisée" (routine preres.f) :
       (f) # si factorisée avec MULT_FRONT :
          (o)   '.VALF  :       OJB      XD  V  R/C     NUM()
          (f)  # si la matrice est non symétrique :
             (o) '.WALF :       OJB      XD  V  R/C     NUM()
       (f) # si factorisée avec LDLT :
          (o)   '.UALF  :       OJB      XD  V  R/C     NUM()
       (f) # si factorisée avec LDLT ou MULT_FRONT :
          (o)   '.DIGS  :       OJB      S   V  R/C

    (f) # si il existe des charges cinématiques :
       (o)   '.CCID'      :       OJB      S   V  I
       (o)   '.CCLL'      :       OJB      S   V  I
       (o)   '.CCVA'      :       OJB      S   V  R/C
       (o)   '.CCII'      :       OJB      S   V  I

/ # Cas particulier du solveur FETI (REFA(5) = 'FETI'):
    (o)   '.LIME'  :  OJB   S  V  K8
    (o)   '.FETM'  :  OJB   S  V  K24 dim = nbsd (nombre de sous-domaines)

    (f) # si au moins un des sous-domaines est flottant :
       (o)   '.FETF'   :  OJB      S   V  I     dim = nbsd
       (o)   '.FETP'   :  OJB      XD  V  I     LONG = nbsdf
                                   nombre de sous-domaines flottants)
       (o)   '.FETR'   :  OJB      XD  V  R       LONG = nbsdf
```

3 Contenu des OJB

3.1 .REFA

.REFA S V K24 dim=11

.REFA(1) = nom du sd_maillage sous-jacent.

.REFA(2) = nom du sd_nume_ddl.

.REFA(3) = ' ' / 'ELIMF' / 'ELIML'
' ' : Il n'existe pas de charges cinématiques.
'ELIMF' : Il existe des charges cinématiques.
Une partie du .VALM a été copiée dans .CCVA
.VALM a été partiellement remplacé par une matrice unité (sur les ddls éliminés)
'ELIML' : Il existe des charges cinématiques.
.CCVA n'a pas encore été créé
.VALM n'a pas encore été recopié dans .CCVA
voir mtmchc.f pour les détails

.REFA(4) = nom de l'OPTION de calcul (ou bien la chaine: '&&MELANGE').

.REFA(5) = ' ' / 'FETI'

.REFA(6) = nom de la structure de données de type sd_feti, si REFA(5)=='FETI',

.REFA(7) = ' ' / nomsolv
nomsolv : nom du solveur à utiliser (par défaut) lors des résolutions.

.REFA(8) = ' ' / 'ASSE' / 'DECT' / 'DECP'
' ' ou 'ASSE' : matrice dans son état initial (non factorisée)
'DECT' : matrice entièrement factorisée
'DEPT' : matrice partiellement factorisée (possible uniquement si LDLT)

.REFA(9) = 'MS' / 'MR'
'MS' : matrice symétrique
'MR' : matrice non-symétrique

.REFA(10) = / 'NOEU': les ddls de la matrice sont portés par des nœuds
/ 'GENE': les ddls de la matrice sont des ddls généralisés.

.REFA(11) = 'MPI_COMPLET' / 'MPI_INCOMPLET' / 'MATR_DISTR'
'MPI_COMPLET' : Les objets .VALM (et .CCVA) sont "complets"
'MPI_INCOMPLET' : Les objets .VALM (et .CCVA) sont "incomplets" du fait d'un calcul MPI distribué. Chaque processeur n'assemble que les éléments finis qui lui sont affectés.
'MATR_DISTR' : Si MATR_DISTRIBUEE='OUI' pour le solveur MUMPS.
Cette option dimensionne au plus juste la matrice assemblée en fonction du nombre d'inconnues présentes sur le processeur courant.

Remarque :
si 'MPI_INCOMPLET', les objets .VALM et .CCVA sont alloués à leur taille normale, mais ils sont partiellement remplis de "0". si 'MATR_DISTR', l'objet .VALM est cette fois d'une taille plus petite que dans le cas 'MPI_INCOMPLET'.

```
DOCU('.REFA') = / 'ASSE'   matrice initiale,
                / 'DECT'   matrice complètement factorisée,
                / 'DECP'   matrice partiellement factorisée.
```

3.2 .VALM

.VALM XD V R/C NUM()

Cet objet contient les valeurs des termes non nuls de la matrice (au format Morse). Il y a 1 (ou 2) élément(s) dans cette collection (2 si la matrice est non symétrique).

Le 1er élément de la collection contient la partie supérieure de la matrice,
le 2ème contient la partie inférieure.

Remarques :

** Pour les matrices non-symétriques,la diagonale est donc stockée 2 fois.*
On convient de n'utiliser que la diagonale stockée dans la partie supérieure (VALM(1)).
** L'arrangement des termes de la matrice dans les blocs est expliqué dans la documentation de la sd_stockage [D4.06.07]*

3.3 .CONL

.CONL S V R dim=neq

neq est le nombre d'équations du système

Cet objet facultatif n'est présent que si il existe au moins un ddl de type 'LAGR' :

```
V(ieq) =  c   si ieq correspond à un DDL nommé 'LAGR',
          1.  sinon.
```

c est le coefficient de conditionnement des ddls de Lagrange.

3.4 .DIGS

.DIGS S V R/C dim=2*neq

neq est le nombre d'équations du système

Cet objet n'est présent que si la matrice a été factorisée par 'LDLT' ou 'MULT_FRONT'

DIGS(1:neq) : valeurs de la diagonale de la matrice initiale
DIGS(neq+1:2*neq) : valeurs de la diagonale de la matrice factorisée

3.5 .LIME

Liste des noms des sd_matr_elem ayant été assemblés pour donner la sd_matr_asse.
Cet objet n'existe que si la sd_matr_asse provient d'un assemblage.
Il vaut mieux éviter de s'en servir.

3.6 .UALF, .VALF, .WALF

Ces collections contiennent les termes de la matrice factorisée.

.UALF contient la factorisée avec LDLT (symétrique ou non-symétrique)
.VALF contient la factorisée avec MULT_FRONT de la partie supérieure
.WALF contient la factorisée avec MULT_FRONT de la partie inférieure

3.7 .CCVA

.CCVA S V R/C

L'objet .CCVA contient les colonnes de la matrice initiale correspondantes aux ddls à éliminer (ceux qui sont imposés par une charge "cinématique"). Plus précisément, on stocke la sous-matrice des termes sur des lignes correspondantes à des DDLs libres et des colonnes à des DDLs imposés.

3.8 .CCII

.CCII S V I

L'objet .CCII a la même structure que l'objet .CCVA. Il contient les indices de lignes des termes stockés dans le .CCVA

3.9 .CCLL

.CCLL S V I dim=3*nelim

nelim est le nombre de ddls "éliminés".

```
.CCLL((i-1)*3+1)    : ieq
.CCLL((i-1)*3+2)    : i1
.CCLL((i-1)*3+3)    : i2
```

ieq est le numéro de l'équation correspondante au i^{ème} ddl éliminé,
i1 est le nombre de termes stockés pour l'équation ieq
i2 est le nombre cumulé de termes stockés pour les équations j < i

3.10 .CCID

.CCID S V I dim=neq+1

.CCID(ieq)= 1 si le ddl ieq éliminé,
0 sinon.
.CCID(neq)= nelim : nombre de ddls éliminés

3.11 .FETM

.FETM S V K24 indirect(*)dim=nbsd (nombre de sous-domaines)

(*): sd_matr_asse non FETI (i.e. FETM(k).REFA(5)('FETI' et pour l'instant imposé à 'MULT_FRONT').

Objet JEVEUX listant les SD sd_matr_asse propres à chaque sous-domaine.

FETM(i) = nom de la SD sd_matr_asse du i^{ème} sous-domaine.

3.12 .FETF

.FETF S V I dim=nbsd

Objet JEVEUX optionnel indiquant le caractère flottant ou non d'un sous-domaine.

.FETF(i)= j (0<j<7) le sous-domaine i comporte j modes de corps rigides.
0 sous-domaine non flottant.

3.13 .FETP

.FETP XD V I LONG=nbsdf

Liste des indices de pivots " quasi-nuls " des sous-domaines flottants.

Soit V=FETP(i),

V(j) : jème indice de pivots du ième sous-domaine flottant.

Le LONMAX de V(i) est égal à FETF(i).

Le nombre de sous-domaines flottants est accessible via l'attribut NUTIOC.

Les sous-domaines flottants sont listés dans le même ordre que les sous-domaines (flottants ou non) de la sd_feti.

3.14 .FETR

.FETR XD V R LONG=nbsdf

Composantes des modes de corps rigides.

Soit V=FETR(i),

V((j-1)*nbddli+k) : $k^{ème}$ composante du $i^{ème}$ sous-domaine flottant.

nbddli, le nombre de DDL du sous-domaine i, est stocké dans sd_feti.FETH(i).

Le nombre de sous-domaines flottants est accessible via l'attribut NUTIOC.

Les sous-domaines flottants sont listés dans le même ordre que les sous-domaines (flottants ou non) de la SD sd_feti.

4 Compléments pour FETI

4.1 Structure de données sd_matr_asse récursive

Dans le cas de la méthode FETI, la structure de données sd_matr_asse est récursive à deux niveaux. Une SD sd_matr_asse " maître ", concernant le domaine global (REFA(5)='FETI'), ne comporte aucun des objets JEVEUX habituels, sauf le .REFA et le .LIME, avec, par contre, en plus des objets spécifiques de la décomposition de domaines : .FETR, FETF, .FETP et .FETM. Ce dernier est un pointeur désignant les SD sd_matr_asse " esclaves " associées à chaque sous-domaines locaux. Ces SD sd_matr_asse locales sont constituées des mêmes objets JEVEUX qu'une sd_matr_asse (.VALE, .CONL...) mono-domaine usuelle. Pour l'instant, l'implémentation de FETI dans Code_Aster présuppose que ces sous-domaines utilisent tous le même solveur linéaire mono-domaine (REFA(5)= 'MULT_FRONT' imposé par défaut, avec donc un .VALF). Cette homogénéité facilite les manipulations des matrices locales.

```
sd_matr_asse "maitre"  ------>   sd_matr_asse "esclave"  (sous-domaine 1)
 (domaine global)      ------>   sd_matr_asse "esclave"  (sous-domaine 2)
   objet .FETM            ...
                         ------>   sd_matr_asse "esclave"  (sous-domaine n)
```

4.2 Règle de nommage

Dans le cas d'un solveur FETI, on a choisi arbitrairement (dans assmam.f) la règle de nommage suivante pour les SD sd_matr_asse esclaves liées chaque à un sous-domaine : nom_de_la_SD_sd_matr_asse_maître(1:11)'F'chaîne_de_caractères_libre(2:8) La chaîne de caractères libre est engendrée par un appel à la routine gcncon.f.

4.2.1 Cas particulier du parallélisme MPI

Lors d'une exécution en mode parallèle MPI, un processeur se voit attribuer un certain nombre de sous-domaines (cf. objets annexes '&FETI.LISTE...' de la structure de données sd_feti[D4.06.21]). La SD sd_matr_asse " maître " est toujours construite, mais son pointeur .FETM ne va désigner que les sous-domaines concernés par le processeur courant : .FETM(jk) sera un K24 valide que si le sous-domaine jk est dans le périmètre du processeur j. Il en va de même pour les objets .FETF, .FETP et .FETR qui ne sont remplis que si nécessaire.

Pour le processeur j :

```
sd_matr_asse "maitre" ------>   "vide"
 (domaine global)     ------>   sd_matr_asse "esclave"  (sous-domaine j1)
   objet .FETM        ------>   sd_matr_asse "esclave"  (sous-domaine j2)
                      ------>   ...
```

Les sous-domaines j1, j2,... sont ceux qui sont concernés par le processeur j