Manuel de Descriptif Informatique
Fascicule D4.06 : Structures liées aux éléments finis
Document D4.06.07

Structures de Données NUME_DDL, NUME_EQUA, STOCKAGE, PROF_CHNO

Résumé :

Ce document décrit les structures de données NUME_DDL, NUME_EQUA, STOCKAGE et PROF_CHNO utilisées pour définir la numérotation des inconnues des systèmes linéaires et le stockage des matrices assemblées et des vecteurs solution et second membre.

1 Généralités

Un NUME_DDL sert à définir la numérotation des inconnues (et des équations) d'un système linéaire. On rappelle que les inconnues d'un tel système sont des CMPS portées par des noeuds du MAILLAGE (ou des noeuds tardifs de LIGREL).
Les matrices que l'on veut décrire sont carrées ; il suffit donc de décrire par exemple leurs lignes.
La numérotation des inconnues d'un système est semblable à celle des CMPS d'un CHAM_NO. D'ailleurs la SD PROF_CHNO décrite dans ce document est celle référencée par la SD CHAM_NO [D4.06.05].
La SD NUME_DDL contient également la description des tableaux de stockage des valeurs des matrices assemblées (SD MATR_ASSE [D4.06.10]). On appelle STOCKAGE cette partie de la structure de données. Les stockages possibles sont le stockage 'MORSE' et 'LIGN_CIEL'.
La SD STOCKAGE décrit comment sont rangés les termes d'une demi-matrice symétrique dans son objet '.VALE'. Lorsqu'une MATR_ASSE n'est pas symétrique, on suppose que son remplissage (ou sa "topologie") reste symétrique. Le STOCKAGE est inchangé mais l'objet '.VALE' est "doublé" : la 1ère partie pour stocker la moitié supérieure, la 2ème partie pour stocker la moitié inférieure.

Attention :

La SD STOCKAGE 'MORSE' est volumineuse (objet .HCOL).

Les relations de dépendance entre ces objets peuvent se représenter par :

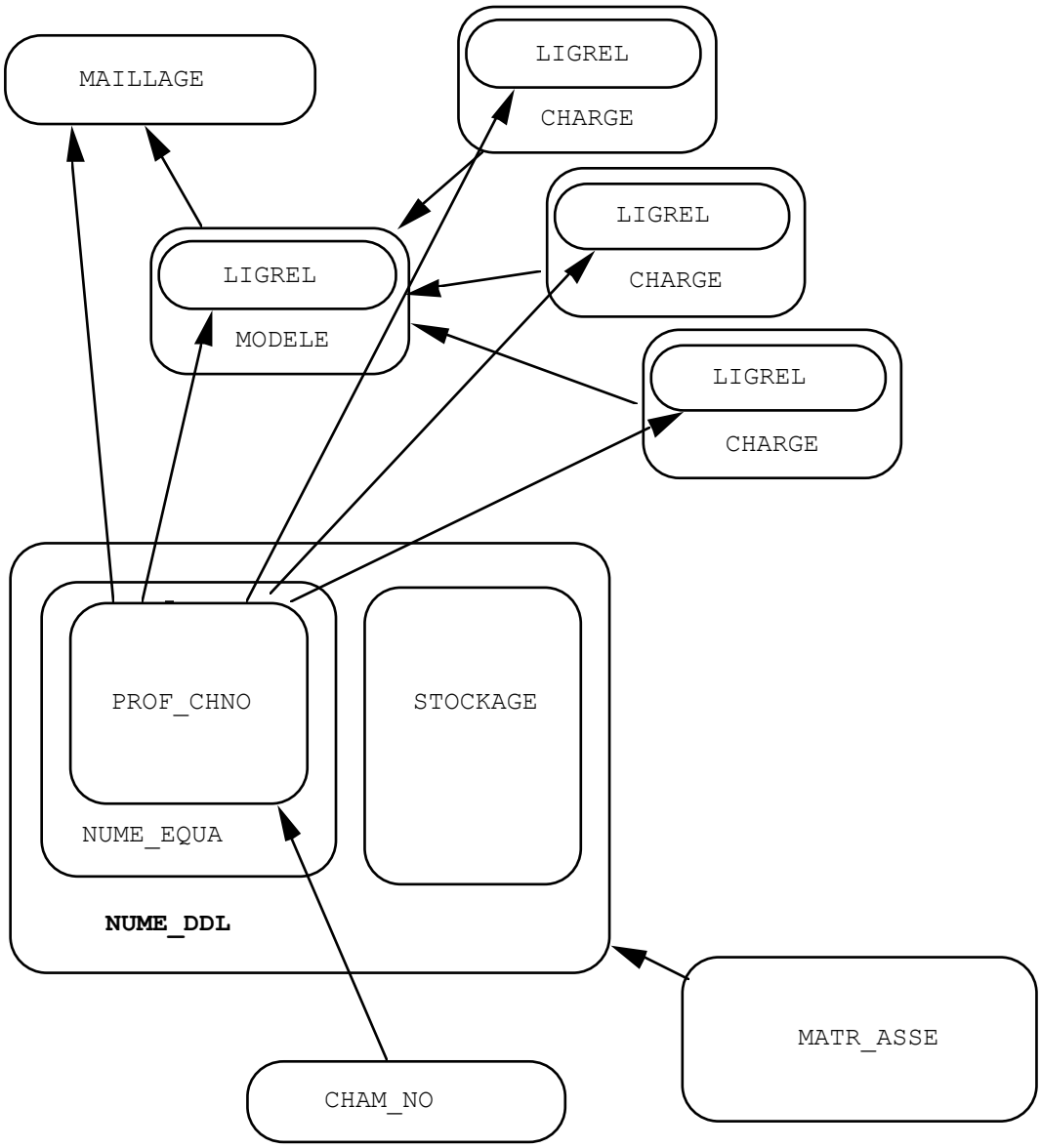


Figure 1-a : Liens des NUME_DDL, NUME_EQUA, STOCKAGE et PROF_CHNO avec les autres structures de données

2 Arborescences

NUME_DDL (K14) ::=record
/ % si solveur LDLT, MULT_FRONT ou GCPC (i.e. NUME.REFN(3)= 'LDLT', 'MULT_FRONT' ou 'GCPC') :
- '.NUME' : NUME_EQUA
- '$VIDE' : STOCKAGE

/ % si solveur MUMPS (i.e. NUME.REFN(3)= 'MUMPS') :
- '.NUME' : NUME_EQUA
- '$VIDE' : STOCKAGE
- '.NSLV' : OJB S V K24

/ % si solveur FETI (i.e. NUME.REFN(3)= 'FETI') :
- '.NUME' : NUME_EQUA
- '.FETN' : OJB S V K24 indirect(*) dim = nbsd (nombre de sous-domaines)
  (*) : NUME_DDL non FETI
  (i.e. FETN(k).NUME.REFN(3)≠'FETI' et pour l'instant imposé à 'MULT_FRONT')

NUME_EQUA (K19) ::=record
- '$VIDE' : PROF_CHNO
- '.DELG' : OJB S V I
- '.NEQU' : OJB S V I
- '.REFN' : OJB S V K24

PROF_CHNO (K19) ::=record
- '.DEEQ' : OJB S V I
- '.LILI' : OJB S N K24
- '.LPRN' : OJB S V I
- '.NUEQ' : OJB S V I
- '.PRNO' : OJB XC V I NOM($.LILI) LONG($.LPRN)

STOCKAGE (K14) ::=record
/ % si solveur 'LDLT'
'.SLCS' : STOC_LCIEL
/ % si solveur 'MULT_FRONT'
'.SMOS' : STOC_MORSE
'.MLTF' : MULT_FRONT
/ % si solveur 'GCPC' ou 'MUMPS'
'.SMOS' : STOC_MORSE

STOC_LCIEL (K19) ::=record
- '.ABLO' : OJB S V I
- '.ADIA' : OJB S V I
- '.DESC' : OJB S V I
- '.HCOL' : OJB S V I
- '.IABL' : OJB S V I
- '.REFE' : OJB S V K24

```
STOC_MORSE  (K19)  ::=record
   ♦ '.ABLO' :  OJBS  V  I
   ♦ '.ADIA' :  OJBS  V  I
   ♦ '.DESC' :  OJBS  V  I
   ♦ '.HCOL' :  OJBS  V  I
   ♦ '.IABL' :  OJBS  V  I
   ♦ '.REFE' :  OJBS  V  K24


MULT_FRONT  (K19)  ::=record
   ♦ '.ADNT' :  OJBS  V  I
   ♦ '.GLOB' :  OJBS  V  I
   ♦ '.LOCL' :  OJBS  V  I
   ♦ '.PNTI' :  OJBS  V  I
```

3 PROF_CHNO

.LILI S N K24
C'est le pointeur de noms de '.PRNO'. Il contient le nom du LIGREL de modèle et ceux des LIGREL à mailles et/ou à nœuds tardifs. Si il s'agit d'un LIGREL à mailles et à nœuds tardifs (DDL_IMPO, LIAISON_DDL…), cela permet d'identifier les nœuds tardifs impliqués dans le PROF_CHNO. Par contre, si il s'agit d'un LIGREL uniquement à mailles tardives (FORCE_NODALE…), il ne pointe vers aucun objet de collection du .PRNO.
Les nœuds d'un PROF_CHNO sont :

- soit les nœuds du maillage ma (concerné par le modèle), sous-jacent au PROF_CHNO,
- soit des nœuds tardifs d'un (ou plusieurs) LIGREL qui s'appuient sur ma.

La collection '.PRNO' contient plusieurs objets :
.PRNO (1) : nœuds du maillage ma
.PRNO (2) : nœuds tardifs du LIGREL dont le nom est dans .LILI(j)
.PRNO (3) : nœuds tardifs du LIGREL dont le nom est dans .LILI(k)
…
On stocke dans .LILI(1) la valeur "&MAILLA"

.LPRN S V I
C'est le pointeur longueur de .PRNO

.PRNO XC V I NOM ($.LILI) LONG ($.LPRN)
Cette collection décrit quels sont les CMPS portées par les nœuds (du maillage ou tardifs) impliqués dans le PROF_CHNO.
Si nec est le nombre d'entiers codés de la grandeur associée au PROF_CHNO,
.PRNO (1) est de longueur (nb_noeuds (ma))*(2+nec)
.PRNO (ili) est de longueur (nb_noeuds_tardifs du ili^{ème} LIGREL)*(2+nec).
Chaque nœud est décrit par 2 entiers et 1 vecteur d'entiers codés de longueur nec que l'on appelle le descripteur-grandeur [D4.06.05].
Prenons l'exemple des nœuds du maillage ma.

```
v = PRNO (1)
v  (ino-1)*(2+nec)+1) = ieq
v  (ino-1)*(2+nec)+2) = nb_cmp

v  (ino-1)*(2+nec)+2+1)
…                                  ] descripteur-grandeur du nœud ino
v  (ino-1)*(2+nec)+2+nec)
```

.nb_cmp est le nombre de CMPS portées par le nœud ino du maillage.
.ieq est l'adresse dans l'objet .NUEQ de la 1ère CMP portée par ino.

Remarques :

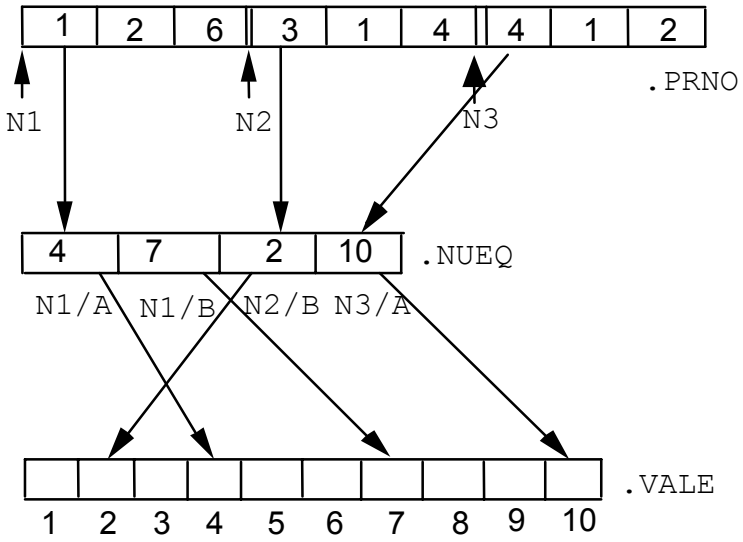
Toutes les CMPS portées par un même nœud sont consécutives dans .NUEQ. C'est pourquoi on ne stocke que l'adresse de la 1ère. Les CMPS sont ordonnées dans l'ordre du catalogue des grandeurs.
Malheureusement, la grandeur associée au .PRNO n'est pas stockée dans le PROF_CHNO.

.NUEQ S V I
C'est un vecteur d'indirection entre l'objet .PRNO et l'objet .VALE des CHAM_NO qui référencent ce PROF_CHNO. Ce vecteur d'indirection permet de s'affranchir de la règle selon laquelle les CMPS d'un nœud se suivent dans l'ordre du catalogue des grandeurs.

Soit par exemple un maillage contenant 3 nœuds : N1, N2, N3
Soit une grandeur gd ayant 2 CMPS : A et B (nec = 1)
Si N1 porte A et B, N2 porte B et N3 porte A.

Remarque :

Pour FETI, on va vérifier que ce vecteur d'indirection est égal à l'identité car sinon cela perturbe l'algorithme de reconstruction du champ solution global (CHAM_NO) à partir des champs solutions locaux (CHAM_NO) à chaque sous-domaine.
Normalement ce cas de figure ne doit se présenter qu'avec les fonctionnalités de la sous-structuration qui est proscrite avec FETI. Mais on ne sait jamais !



.DEEQ S V I DIM = 2*neq si neq est le nombre d'équations du PROF_CHNO
C'est un vecteur "inverse" de .PRNO qui décrit (partiellement) les équations.
Si nueq est un numéro d'équation (i.e. adresse dans l'objet .VALE).

```
V ((nueq-1)*2+1) : ino
V ((nueq-1)*2+2) : icmp
```

- Si ino > 0 et icmp > 0
  nueq est l'équation associée à la icmp^{ième} CMP portée par le nœud ino du maillage.
- Si ino > 0 et icmp < 0
  nueq est l'une des 2 équations qui dualisent le blocage de la icmp^{ième} CMP du nœud ino du maillage.
- Si ino = 0 et icmp = 0
  nueq est une équation de dualisation d'une relation linéaire entre plusieurs CMP.

4 NUME_EQUA

.REFN S V K24 dim = 4
.REFN(1) = nom du maillage sous-jacent au NUME_DDL.
.REFN(2) = nom de la grandeur simple associée NOMGDS TEMP R, DEPL R, PRES_C…
.REFN(3) = type de résolution du solveur linéaire : 'LDLT', 'GCPC', 'MULT_FRONT' 'MUMPS' ou 'FETI' (information provenant de SOLVEUR.SLVK(1)).
.REFN(4) = nom de la structure de données de type SD_FETI (information provenant de SOLVEUR.SLVK(6)).

.NEQU NEQU(1) nombre total d'équations.
NEQU(2) inutilisé

.DELG S V I dim = neq
cet objet décrit (un peu) les équations du type "Lagrange".
DELG(ieq) = / -1
si l'équation ieq correspond à la CMP 'LAGR' porté par le 2ème nœud d'une maille SEG3 portant un élément de Lagrange (Lagrange 1)
DELG(ieq) = / -2
si l'équation ieq correspond à la CMP 'LAGR' portée par le 3ème nœud d'une maille SEG3 portant un élément de Lagrange (Lagrange 2)
DELG(ieq) = / 0 sinon

.FETN S V K24 indirect(*) dim = nbsd (nombre de sous-domaines)
(*) : NUME_DDL non FETI
(i.e. FETN(k).NUME.REFN(3)≠'FETI' et pour l'instant imposé à 'MULT_FRONT')
Objet JEVEUX optionnel (présent uniquement pour domaine global si FETI, puis absent pour chaque sous-domaine) listant les SD NUME_DDL propres à chaque sous-domaine.

5 Complément pour MUMPS

.NSLV S V K24 dim = 1
.NSLV(1) = nom de la SD SOLVEUR.

6 Compléments pour FETI

6.1 Structure de données NUME_DDL récursive

Dans le cas de la méthode FETI, la structure de données NUME_DDL est récursive à deux niveaux. Une SD NUME_DDL « maître », concernant le domaine global (.NUME.REFN(3)='FETI'), comporte les objets JEVEUX habituels complétés par un objet spécifique de la décomposition de domaines : le .FETN.
C'est en fait un pointeur désignant les SD NUME_DDL « esclaves » associées à chaque sous-domaines locaux. Ces SD NUME_DDL locales sont constituées des mêmes objets JEVEUX qu'un NUME_DDL mono-domaine usuel.
Pour l'instant, l'implémentation de FETI dans *Code_Aster* présuppose que ces sous-domaines utilisent tous le même solveur linéaire mono-domaine (.NUME.REFN(3)='MULT_FRONT' imposé par défaut). Cette homogénéité facilite les manipulations des matrices et seconds membres locaux.
Bien sûr, lors de la factorisation symbolique, on créé les structures de données STOCKAGE des sous-domaines mais pas celle du domaine global. Seul le NUME_EQUA de ce dernier voit le jour, pour pouvoir manipuler un champ global solution et… pointer vers les champs locaux.

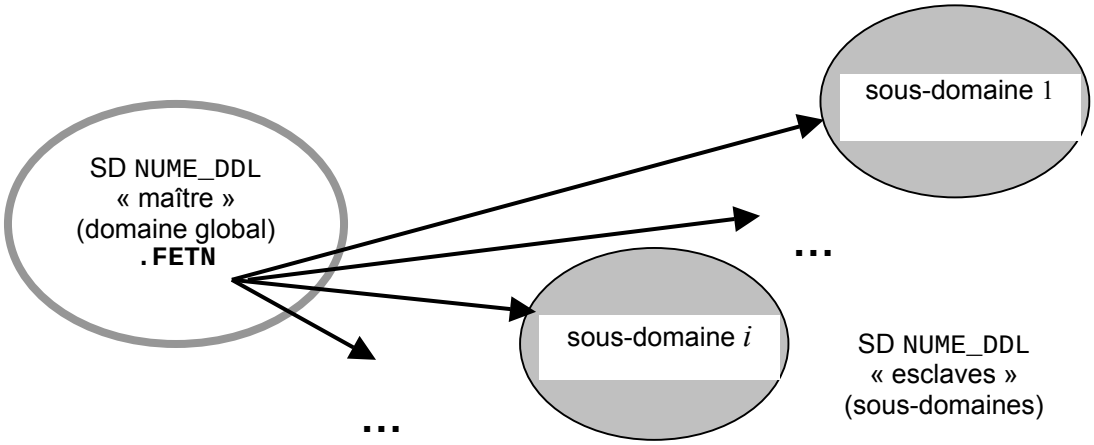


Figure 6.1-a : Structure de données NUME_DDL récursive si solveur FETI

6.2 Règle de nommage

Dans le cas du solveur FETI, on a choisi la règle de nommage suivante pour la SD NUME_DDL esclave liée à un sous-domaine j :

nom_de_la_SD_NUME_DDL_maître(1:6)//'F'//chaîne_de_caractères_libre(2:8)

La chaîne de caractères est engendrée par un appel à la routine GCNCON.

D'autre part, les pré-requis des routines de constitution des NUME_DDL (esclaves) imposent la création de LIGRELs temporaires nommés

'&F'//'*chaîne_de_caractères_libre(3:8)*/'.MODELE '

6.3 Vérification du .NUEQ

Dans le cas du solveur FETI, on vérifie que le vecteur d'indirection .NUEQ du NUME_DDL maître soit égal à l'identité car cette hypothèse facilite l'opération de reconstruction du champ résultat global à partir des champs résultats locaux. Normalement, ce cas de figure ne doit se produire que pour la sous-structuration qui est de toute façon contre-indiquée, pour d'autres raisons, avec FETI.

6.4 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 NUME_DDL « maître » est toujours construite, mais son pointeur .FETN ne va désigner que les sous-domaines concernés par le processeur courant : .FETN(j_k) sera un K24 valide que si le sous-domaine j_k est dans le périmètre du processeur j.

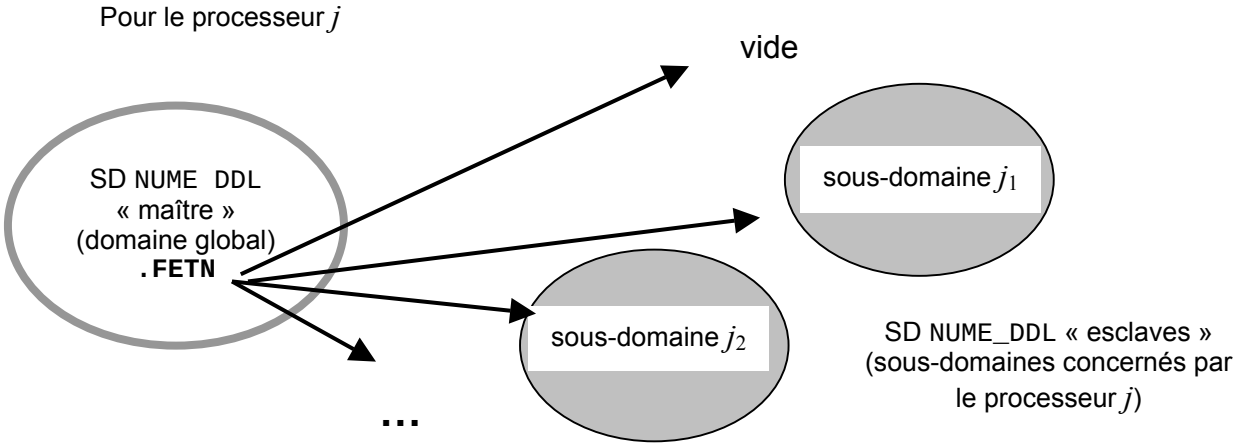
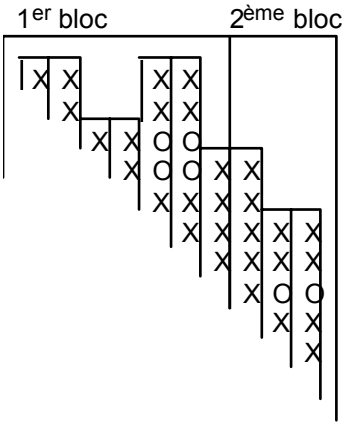


Figure 6.4-a : Structure de données NUME_DDL récursive si solveur FETI et parallélisme MPI

7 STOC_LIGN_CIEL

.REFE	(1)	nom de la numérotation supportant ce stockage.
.DESC	(1)	nombre d'équations : neq
	(2)	taille des blocs de la matrice : t_bloc
	(3)	nombre de blocs nécessaire au stockage des valeurs de la matrice : n_bloc
	(4)	hauteur maximum des colonnes de la matrice
.HCOL		S V I dim = neq
.HCOL	(i)	hauteur de la $i^{ème}$ colonne
.ADIA		S V I dim = neq
.ADIA	(i)	adresse du terme diagonal de la $i^{ème}$ colonne dans son bloc
.ABLO		S V I dim = n_bloc + 1
.ABLO	(1)	0
	(K+1)	numéro de la dernière colonne du bloc K . remarque : une colonne ne peut appartenir qu'à un seul bloc
.IABL		S V I dim = neq
.IABL	(i)	numéro du bloc qui contient la $i^{ème}$ colonne de la matrice

Exemple :




```
HCOL(1)     =     1
HCOL(2)     =     2
HCOL(3)     =     1
HCOL(4)     =     2
HCOL(5)     =     5
HCOL(6)     =     6
HCOL(7)     =     5
HCOL(8)     =     6
HCOL(9)     =     5
HCOL(10)    =     6
```

```
ADIA(1)   =  1          ADIA(6)    =  17
ADIA(2)   =  3          ADIA(7)    =  22
ADIA(3)   =  4          ADIA(8)    =  6
ADIA(4)   =  6          ADIA(9)    =  11
ADIA(5)   =  11         ADIA(10)   =  17

ABLO(1)   =  0          IABL(1 à 7)   =  1
ABLO(2)   =  7          IABL(7 à 10)  =  2
ABLO(3)   =  10
```

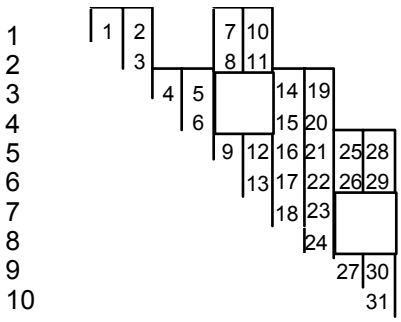
8 STOC_MORSE

.REFE	(1)	nom de la numérotation supportant ce stockage.
.DESC	(1)	nombre d'équations : neq
	(2)	taille du bloc de la matrice : t_bloc
	(3)	nombre de bloc : 1
	(4)	nombre de termes stockés dans la matrice : n_termes
.HCOL		S V I dim = n_termes
.HCOL	(i)	numéro de ligne du $i^{ème}$ terme stocké
.ADIA		S V I dim = neq
.ADIA	(i)	adresse du terme diagonal de la $i^{ème}$ colonne dans le bloc Il faut donc que tous les termes diagonaux soient stockés dans la matrice
.ABLO	(1)	0
	(2)	neq
.IABL		S V I dim = neq
.IABL	(i)	1

Remarque :

Aujourd'hui (août 2004), le remplissage de la matrice (objet .HCOL) est tel que tous les DDLs portés par un nœud sont connectés à tous les DDLs portés par les autres nœuds jouxtant ce nœud via un élément fini. La matrice est donc formée de petits rectangles pleins correspondant à la connectivité de ces nœuds.

Exemple :




```
HCOL(1)      =      1
HCOL(2)      =      1
HCOL(3)      =      2      ...
HCOL(4)      =      3      HCOL(28)     =      5
HCOL(5)      =      3      HCOL(29)     =      6
HCOL(6)      =      4      HCOL(30)     =      9
HCOL(7)      =      1      HCOL(31)     =      10
HCOL(8)      =      2
HCOL(9)      =      5
```

```
ABLO(1)          =  0
ABLO(2)          =  10

IABL(1 à 10) =  1

ADIA(1) = 1      ADIA(6)   =  13
ADIA(2) = 3      ADIA(7)   =  18
ADIA(3) = 4      ADIA(8)   =  24
ADIA(4) = 6      ADIA(9)   =  27
ADIA(5) = 9      ADIA(10)  =  31
```

9 MULT_FRONT

Soit lgind, la somme du nombre des voisins des super-nœuds.

.GLOB : S V I
dimension = lgind
Ce vecteur donne l'ensemble des voisins des super-nœuds

.LOCL : S V I
dimension = lgind
Ce vecteur établit pour les numéros de lignes des supernœuds, la correspondance entre la numérotation locale du fils et la numérotation locale du père.

.ADNT : S V I
dimension = taille de la matrice initiale (morse)
C'est le vecteur des adresses des termes initiaux dans la matrice factorisée.

.PNTI : S V I
dimension = 19*neq+10
Cumul dans un même vecteur d'une suite de tables de travail.

10 Exemples

Un exemple de NUME_DDL associé aux 4 solveurs linéaires est donné en :

'LDLT'	[D4.06.10 §5.2]
'MULT_FRONT'	[D4.06.10 §5.3]
'GCPC'	[D4.06.10 §5.4]
'FETI'	[D4.06.10 §5.5]

Page laissée intentionnellement blanche.