Data format sd_nume_dll, sd_nume_equa, sd_stockage and sd_prof_chno

Summarized:

This document describes data structures NUME_DDL , NUME_EQUA , STOCKAGE and PROF_CHNO used to define the classification of the unknowns of the linear systems and the storage of the assembled matrixes and the vectors solution and second member.

Contents

1 General information

a NUME_DDL are used to define the classification of the unknowns (and the equations) of a linear system. It is pointed out that the unknowns of such a system are CMPS carried by nodes of MAILLAGE (or late nodes of LIGREL).

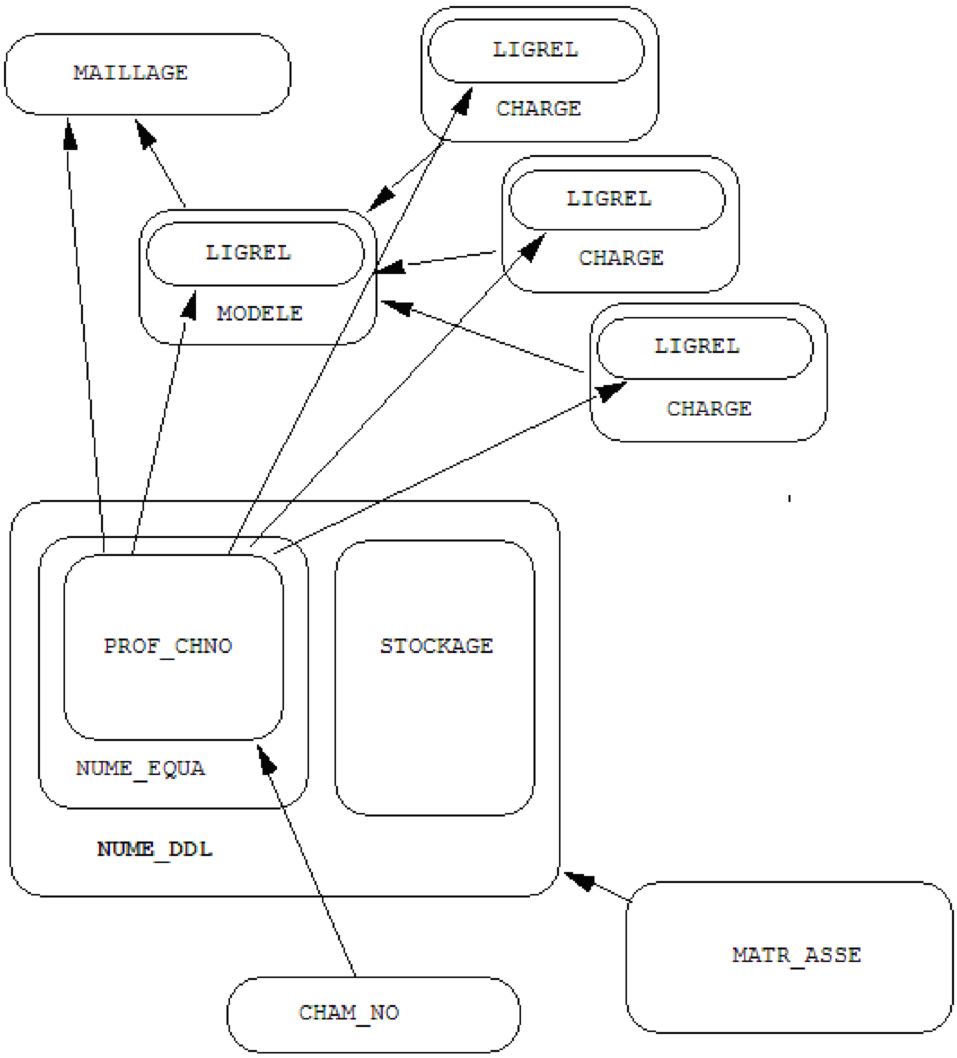
The matrixes which one wants to describe are square, hollow, symmetric or not.

When a matrix is NON-symmetric, its hollow structure is symmetric.

The classification of the unknown of a system is similar to that of the CMPS of a CHAM_NO . Moreover SD PROF_CHNO described in this document is that referred by SD CHAM_NO [D4.06.05 - Data structure field].

SD NUME_DDL also contains the description of the tables of storage of the values of the assembled matrixes [D4.06.10 - Data format sd_matr_asse]. One calls STOCKAGE this part of data structure.

The relations of dependence between these objects can be represented by:



Appear 1-a: Restrains of the NUME_DDL, NUME_EQUA, STOCKAGE and PROF_CHNO with the other data structures

2 Tree structures

```
NUME_DDL (K14):: =record

    ♦ ".NUMÉRIQUE"   :    NUME_EQUA
    ♦ ".NUML"   :    NUML_EQUA
    ♦ ".NSLV"   :    OJB S  V  K24  long=1

    / % if solver not FETI ("MULT_FRONT", "MUMPS", "PCG",…) :
        ♦ "$VIDE"   :    STOCKAGE

    / % so FETI solver (i.e. NUMÉRIQUE.REFN (3) = "FETI"): ♦
        ". FETN":   OJB    S  V  K24  indirect (*) dim = nbsd (or
                                        nbsd = many subdomains) (*
                                        ): NUME_DDL not FETI (i.e.
                                  FETN ( K) .NUMÉRIQUE.REFN (3)? "FETI" and for the time
                                     imposed on "MULT_FRONT  ") NUMÉRIQUE_

EQUA (K19) :: =record ♦ "

        $VIDE  ": PROFESSOR     _     CHNO ♦ "
        .NEQU  ": OJB     S     V I  long  =    2 ♦ "
        .REFN  ": OJB     S     V K24   long=  4 ♦ "
        .DELG  ": OJB     S     V I  NUML  _


EQUA (K19) :: =record ♦ "

        .PRNO  ": OJB     XC     V I ♦   "
        .NEQU  ": OJB     S     V I  long  =    2 ♦ "
        .DELG  ": OJB     S     V I  ♦  "
        .NUEQ  ": OJB     S     V I  ♦  "
        .NULG  ": OJB     S     V I  ♦  "
        .NUGL  ": OJB     S     V I  PROFESSOR  _

CHNO (K19) :: =record ♦ "

        .DEEQ  ": OJB   S    V I  ♦  "
        .LILI  ": OJB   S    N K24  ♦  "
        .NUEQ  ": OJB   S    V I  ♦  "
        .PRNO  ": OJB   XC    V I NOM  ($  .LILI) STOCKAGE

(K14) :: =record ♦ "

    .SMOS   ": STOC     _MORSE #   to describe the initial hollow matrix | %
    if factorized by "LDLT " ".SLCS
       ": STOC     _LCIEL #   to describe the factorized matrix | %
    if factorized by "MULT_FRONT " ".MLTF
       ": MULT_FRONT     #   to describe factorized matrix STOC_
```

LCIEL (K19) :: =record ♦ ‶

.SCBL ″: OJB S V I ♦ ‶
.SCDI ″: OJB S V I ♦ ‶
.SCDE ″: OJB S V I ♦ ‶
.SCHC ″: OJB S V I ♦ ‶
.SCIB ″: OJB S V I STOC _

MORSE (K19) :: =record ♦ ‶

.SMDI ″: OJB S V I ♦ ‶
.SMDE ″: OJB S V I ♦ ‶
.SMHC ″: OJB S V I MULT_FRONT

(K19) :: =record ♦ ‶

.ADNT ″: OJB S V I ♦ ‶
.GLOB ″: OJB S V I ♦ ‶
.LOCL ″: OJB S V I ◊ ‶
.PNTI ″: OJB S V I NUMERICAL _

3 DDL .NSLV

S V K24 dim = 1 NSLV (

- 1) name of the sd_solvor who will be used by default. NUME_

4 EQUA .REFN

S V K24 dim = 4 .REFN
(1) = name of the mesh subjacent with the NUME_DDL . .REFN
(2) = name of associated simple quantity NOMGDS TEMP_R , DEPL_R , NEAR _ C…. REFN
(3) = linear solver by default: “LDLT ″, “ PCG ″, “ MULT_FRONT ″ ‶ MUMPS ″ or “ FETI ″ (information coming solver .SLVK (1)). .REFN
(4) = name of data structure of SD_FETI type (information coming solver .SLVK (6)). Note:

REFN (
3) is redundant (or contradictory?) with. NSLV (1). .NEQU

NEQU (
1) nombre total of equations. NEQU (
2) unutilised .DELG

S V I dim = neq this object
described (a little) the equations of the type “Lagrange”. DELG (
ieq) =/-1 if
the equation ieq corresponds to CMP “LAGR ″ carried by the 2nd node of a mesh SEG3 carrying an element from Lagrange (Lagrange 1) DELG (
ieq) =/-2 if
the equation ieq corresponds to CMP “LAGR ″ carried by the 3rd node of a mesh SEG3 carrying an element of Lagrange (Lagrange 2) DELG (

ieq) =/0 if not NUML_

5 EQUA This

data structure is built only if the solver used is MUMPS and that the user activated option MATR_ DISTRIBUTES = ' OUI'. .PRNO

This object

is identical (in its architecture) to. PRNO of a PROF_ CHNO. The difference lies in the fact that the classification of the ddls is local for the processor considered. .NEQU

NEQU (

- 1) local number of equations. NEQU (
- 2) unutilised .DELG

S V I dim = neq local This object

is similar to the object. DELG of a NUME_ EQUA. .NUEQ

S V I dim = neq local It is

a vector of indirection between object .PRNO and object .VALE of the CHAM_ NO cf .NUEQ of a NUME_ EQUA .NULG

S V I It is

a vector of indirection which makes it possible to pass from the local classification of the d.o.f. (of the .NUML) to total classification (of the .NUMÉRIQUE) .NUGL

S V I It is

a vector of indirection which makes it possible to pass from the total classification of the d.o.f. (of the .NUMÉRIQUE) to local classification (of the .NUML) PROFESSEUR_

6 CHNO .LILI

S N K24 It is

a pointer of names which gives access the elements of the collection ". PRNO ". By convention

, one stores in. LILI (1) the value "& NETTED ", which wants to say that L" object PRNO (1) informs about the ddls carried by the nodes of the mesh subjacent with PROF_ CHNO. The other

objects of collection .PRNO (when they exist) inform about the ddls carried by the nodes which N" do not belong to the mesh (late nodes). This late nodes are defined in LIGREL . A each LIGREL nomlig corresponds an object .PRNO (ili). The number of the LIGREL (ili) is that corresponding to nomlig in the pointer of name .LILI It contains

the name of the LIGREL of model and those of the LIGREL to meshes and/or late nodes. If it is about a LIGREL to meshes and with late nodes (DDL_IMPO , LIAISON_DDL ...), that makes it possible to identify the late nodes implied in PROF_ CHNO. On the other hand , if it is about a

LIGREL only to meshes late (FORCE_NODALE …), it does not point towards any object of collection of. PRNO . The nodes

of a PROF_ CHNO are : either

- nodes of the mesh my (concerned with the model), subjacent with PROF_ CHNO, or
- of the late nodes of one (or several) LIGREL which S "lean on my. The collection

". PRNO " contains several objects: . PRNO

(1): nodes of the mesh my. PRNO
(2): late nodes of the LIGREL whose name is in. LILI (2). PRNO
(3): late nodes of the LIGREL whose name is in. LILI (3)…. PRNO

XC V I NUM () This

collection describes which are the CMPS carried by the nodes (of the mesh or late) implied in PROF_ CHNO. If nec
is the number D" whole coded quantity associated with PROF_ CHNO. PRNO

- (1) is length (nb_noeuds (my))* (2+nec). PRNO
- (ili) is length (nb_ nodes _tardifs ili ème LIGREL) * (2+nec). Each

node is described by 2 integers and 1 vector of coded integers length nec that L "one calls the descriptor-quantity [D4.06.05 - Data structure field]. Let us take

L" example of the nodes of the mesh my. nb_ cmp

is the number of CMPS carried by the node ino of the mesh . ieq is
L "addresses in L" object. NUEQ of the 1st CMP carried by ino. 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) | …
| Descriptor                              - quantity of the node ino v (ino
-  1) * (2+nec) +2+nec | < -    Note:
```

All

the CMPS carried by the same node are consecutive in. NUEQ . This is why one does not store that the address of 1st . The CMPS are ordered in the order of the catalog of quantities. Unfortunately
, quantity associated with. PRNO is not stored in PROF_ CHNO. .NUEQ

S V I It is

a vector of indirection between the object. PRNO and the object. VALE of the CHAM_NO which refer this PROF_ CHNO. In
the practice, this vector is almost always identity (NUEQ (I) =I). This initialization is made in the routine nueffe.f during the creation of a NUME_DDL . This
vector is modified when one creates macr_elem _stat with an aim of separating the internal ddls from the external ddls of the macro-element. One can then factorize partially the matrixes (the part corresponding to the internal ddls. Note:

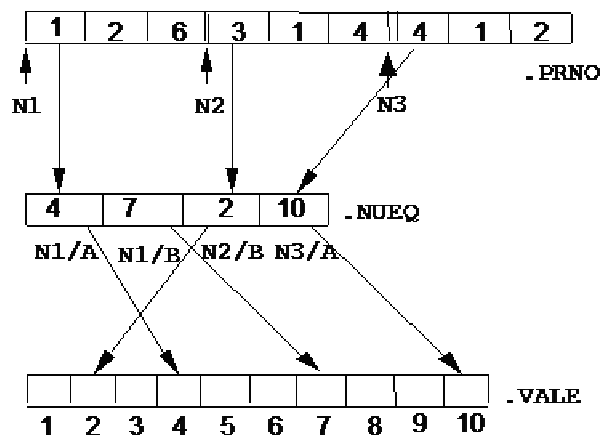
If the solver

- is MULT_FRONT , it is checked that the vector corresponds to the identity (routine mltpre.f). For FETI
- *, one will check that this vector of indirection is equal to the identity because if not that disturbs the algorithm of rebuilding of the total field solution (CHAM_NO) from the fields solutions buildings (CHAM_NO) with each under field. Normally this case should arise only with the functionalities of the substructuring which is proscribed with FETI. But one never knows! This vector*

of indirection would theoretically make it possible to be freed from the rule according to which the CMPS of a node are followed in the order of the catalog of quantities as one can see it on the following example. That is to say a mesh

containing 3 nodes: N1, N2, N 3 Is a quantity
Gd having 2 CMPS : A and B (nec = 1) If N1 carries

A and B, N2 carries B and N3 carries A. .DEEQ S V I



DIM = 2*neq if neq is

the number of equations of the PROF_CHNO C "is

a vector "reverses". PRNO which describes (partially) equations. If nueq is
a number D" equation (i.e addresses in the object. VALE). V ((nueq – 1) *2+

1) : ino V ((nueq-1) *2+
2) : ICMP If ino > 0 and

- ICMP > 0 nueq is L "equation
  associated with ICMP ième CMP carried by the node ino with the mesh . If ino > 0 and

- ICMP < 0 nueq is L " one
  of the 2 equations which dualisent the blocking of ICMP ième CMP of the node ino of the mesh . If ino = 0 and

- ICMP = 0 nueq is an equation
  of dualisation of a linear relation between several CMP. Complements for

7 FETI .FETN S V K24

indirect (*) dim = nbsd (many subdomains) (*): NUME_DDL

not FETI (i.e. FETN (K) .NUMÉRIQUE
.REFN (3)? "FETI" and for the time imposed on "MULT_FRONT") optional JEVEUX object

(present only for total field if FETI, then absent for each subdomain) listing specific SD NUME_DDL to each subdomain. Data format

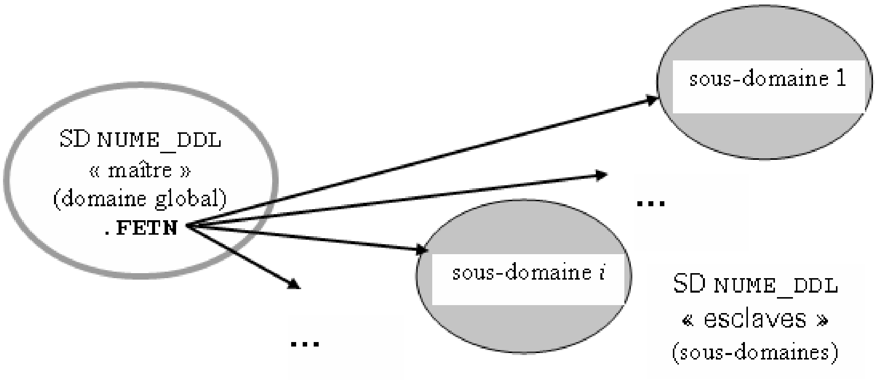
7.1 NUME_DDL recursive In the case of

method FETI, data structure NUME_DDL is recursive on two levels. A SD NUME_DDL "Master ", concerning the total field (. NUMÉRIQUE.REFN (3) = ' FETI "), comprises the usual JEVEUX objects supplemented by a specific object of the decomposition of fields: . FETN. C" in fact a pointer

indicating is SD NUME_DDL "slaves " associated with each local subdomains. These SD NUME_DDL local are made up by the same JEVEUX objects as a NUMERICAL _DDL usual mono-field . For time,

the implementation of FETI in Code_Aster presupposes *that these* subdomains use all the same linear solver mono-field (. NUMÉRIQUE.REFN (3) = ` MULT _FRONT' imposed by default). This homogeneity facilitates handling of the matrixes and second local members. Of course, during

factorization symbolic system, one created data structures STOCKAGE of under fields but not that of the total field. Only the NUME_EQUA of this last comes out , to be able to handle a global field solution and… to point towards the local fields. Appear 7.1-a: Data format



7.1-aNUME_DDL recursive so FETI solver Rule of naming

7.2 In the case of FETI solver

, one chose the following rule of naming for SD NUME_DDL slave related to a subdomain J: nom_de_la_SD_ *NUMERICAL*

_DDL_maître (1:6)/"F" //chaîne_de_caractères_libre (2:8) the character string

is generated by a call to routine GCNCON. In addition , them pre-necessary of the routines of constitution of the NUME_DDL (slaves) impose the creation of LIGRELs temporary named "&F"/"character string

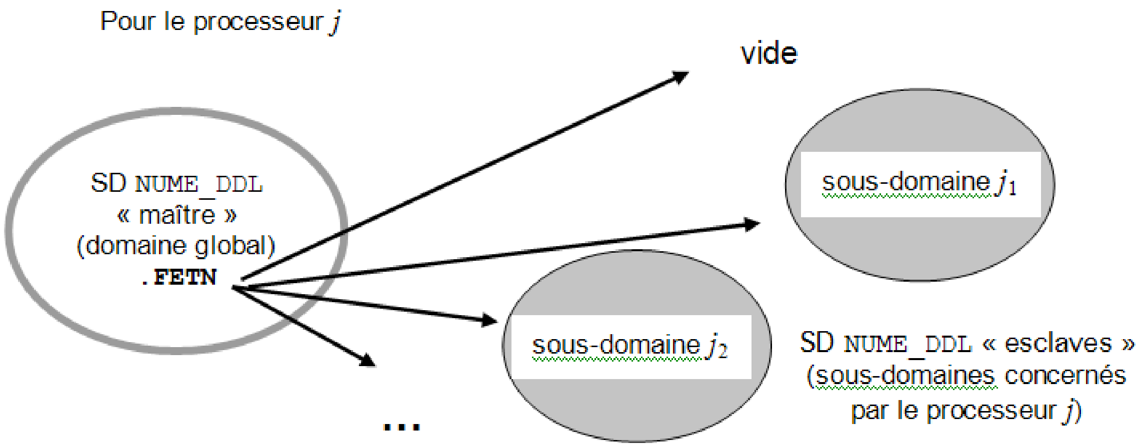
_de_caractères_libre (3:8)/" .MODELE "Checking of

7.3 the .NUEQ In the case of FETI solver

, one checks that the vector D" indirection. Main NUEQ of the NUME_DDL is equal to the identity because this assumption facilitates the operation of rebuilding of the field result total starting from the fields local results. Normally, this case must occur only for the substructuring which is in any case contra-indicated, for other reasons, with FETI. Typical case of

7.4 parallelism MPI During an execution

in parallel mode MPI, a processor is seen allotting a certain number of subdomains (additional cf objects "&FETI.LISTE… " of data structure SD_FETI (D4.06.21 – Data format SD_FETI). SD NUME_DDL "Master " is always built, but its pointer. FETN will indicate only the subdomains concerned with the processor running: . FETN (J K) will be one K24 $_{valid}$ that if the subdomain J K is in the perimeter of the processor J. Appear 7.4-a : Data format



7.4-aNUME_DDL recursive so FETI solver and parallelism MPI STOCKAGE the hollow

8 matrixes

of Code_Aster have a whole a symmetric structure: if has (I, J) exists, then has (J, I) also exists . For a hollow

symmetric matrix, one stores only his "higher" part (has (I, J) for J ≥ I) . Data structure STOCKAGE describes how the non-zero terms of the higher part of matrix are stored. For an asymmetric

matrix, one stores the upper part and the lower part. As these 2 parts have same structure (except for a transposition), the storage of the upper part is enough. Note: Although

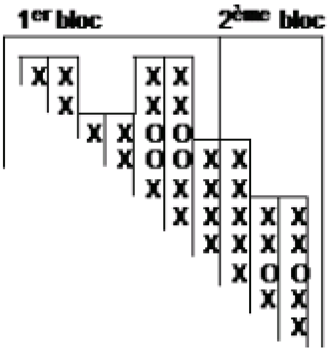
theoretically

, the non-zero terms of a matrix can be had unspecified way, the filling of the matrixes of Code_Aster is such that all DDLs carried by a node are connected to all DDLs carried by the other nodes being next to this node via a finite element. The matrix is thus formed small rectangles full corresponding to connectivity with the nodes of the model. In particular, there exist small blocks on all the diagonal of the matrix. STOC_LIGN_CIEL .SCDE

8.1 (1) (2) (

3) (4	) many equations	: neq size of the blocks of the matrix: t_bloc number of blocks necessary to the storage of the values of the matrix: n_bloc maximum height of the columns of matrix .SCHC S V I dim =
neq.		SCHC (I) height
of	ième	column .SCDI S V I dim =
neq.		SCDI (I) S
addresses		diagonal term of the ième column in its block .SCBL V I dim =
n_bloc		+ 1 .SCBL (1) (K+1)
0 number	of the last	column of the block K. note: a column can belong only to one block .SCIB S V I dim =
neq.		SCIB (I) number
of the block	which	contains I ème column of the matrix Example: SCDI (1)

= 1 SCDI




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

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

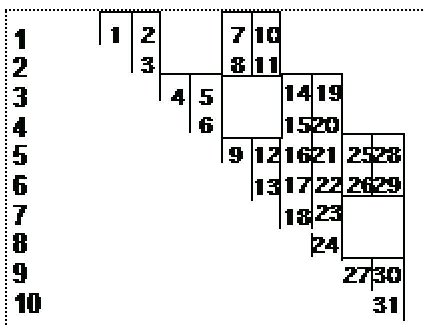
(10) = 6 SCBL (1) = 0 SCIB

(1 to 7) = 1 SCBL (2) = 7 SCIB
(7 to 10) = 2 SCBL (3) = 10 STOC
_MORSE .SMDE

8.2 (1) (2)

(3) many	equations	: neq many terms stored in the half-matrix: n_termes number of block: 1 .SMHC S V I dim =
n_termes		.SMHC (I) issue of line
		of the ième term stored .SMDI S V I dim =
neq.		SMDI (I) addresses
diagonal		term of the ième column in the block It is necessary thus that all the diagonal terms are stored in the matrix Example: SMDI (1)

= 1 SMDI



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

(2) = 1 SMHC
(3) = 2 SMHC
(4) = 3 SMHC
(5) = 3 SMHC
(6) = 4 SMHC
(7) = 1 SMHC
(8) = 2 SMHC
(9) = 5 …
SMHC (28) = 5
SMHC
(29) = 6 SMHC
(30) = 9 SMHC
(31) = 10 MULT_FRONT

Are

8.3 lgind,

the sum amongst neighbors of the super-nodes. .GLOB: S V I4 dimension

= lgind
This vector gives
all the neighbors of super-nodes .LOCL: S V I4 dimension

= lgind
This vector establishes
for the issues of line of the super-nodes, the correspondence between the local classification of the wires and the local classification of the father. .ADNT: S V I dimension

= size
of the initial matrix (Morse) It is the vector
of the addresses of the initial terms in the factorized matrix. .PNTI: S V I dimension

= 19*neq
+10 Office plurality in the same
vector of a succession of work tables. Examples an example

9 of NUMERICAL

_DDL associated with the 4 linear solvers (`LDLT `, ` MULT_FRONT ` , ` PCG ` , “ FETI ”) is given in the document [D4.06.10 - Data format sd_matr_asse].