Data format sd_matr_asse

Summarized:

This document describes data structures sd_matr_asse : hollow matrixes.

Contents

1 General information

the objects of the type sd_matr_asse represent the square assembled matrixes (within the meaning of the finite elements). It is in general of large objects. These matrixes are hollow, which explains why their structure is not simply a square table.

One sd_matr_asse can come from an assembly of sd_matr_elem or of a linear combination of others sd_matr_asse.

The tables describing the storage of sd_matr_asse are in the structure sd_stockage of a sd_nume_ddl [D4.06.07].

There exists sd_matr_asse symmetric and of sd_matr_asse NON-symmetric. But it is supposed that the topology of the matrix is always symmetric. I.e. that the non-zero terms are laid out symmetrically compared to the diagonal.

2 Tree structures

```
sd_matr_asse  (K19)  :: =record

(O)   ".REFA"          :       OJB      S   V  K24

/# general Case (solver /= FETI i.e. REFA (5)/= "FETI"): (O
   ) ".   VALM":    OJB       XD      V  R  C NUM   () nbobj=1/2 (F

   ) # if there exist ddls of Lagrange: (O
      ) ".   CONL": OJB       S       V   R  (F

   ) #  if the sd _matr_asse comes from the assembly of elementary matrixes: (O
      ) ".   FILE": OJB       S       V   K8  (F

   ) #  if the sd _matr_asse "was factorized" (routine preres.f ): (F)
      # if  factorized with MULT_FRONT : (O)
         ".VALF   : OJB        XD V      R  C  NUM ()      (F)
         # if  the matrix is asymmetric: (O)"
            .WALF: OJB       XD V      R  C  NUM ()      (F)
      # if  factorized with LDLT : (O)
         ".UALF   : OJB        XD V      R  C  NUM ()      (F)
      # if  factorized with LDLT or MULT_FRONT : (O)
         " .DIGS   : OJB        S V      R   C (  F) #

   if  there  exist kinematical loads: (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   /#  Casparticulier

of FETI solver (REFA (5) = "FETI"): (O)
   ".LIME   ": OJB   S V   K8  (O  )
   ".FETM   ": OJB   S V   K24  dim  = nbsd (many subdomains) (F) #

   so  at least  one of the subdomains is floating: (O)
      ".FETF   ": OJB    S V      I   dim  =     nbsd (O)
      ".FETP   ": OJB    XD V      I  LONG      = nbsdf many
                                    floating subdomains) (O)
      ".FETR   ": OJB    XD V      R  LONG       = nbsdf Contained
```

3 OJB .REFA

3.1 .REFAS

V K24 dim=11 .REFA

(1) = name of the subjacent sd_maillage..REFA

(2) = name of the sd_nume _ddl. .REFA

(3) = "" /"ELIMF "/"ELIML """:
It does not exist of kinematical loads. "ELIMF
": There exist kinematical loads. Part
of the .VALM was copied in .CCVA .VALM
was partially replaced by a matrix unit (on
the eliminated ddls) "ELIML
": There exist kinematical loads. .CCVA
was not created yet .VALM
was not recopied yet in .CCVA to see
mtmchc.f for details .REFA (4)

= name of the Computation option (or the character string: "&&MELANGE "). .REFA (5)

= ""/"FETI ". REFA (6)

= name of data structure of the sd_feti type, if REFA (5) = = ' FETI", .REFA (7)

=" "/nomsolv nomsolv
: name of the solver during to use (by default) the resolutions. .REFA (8)

=" "/ "ASSE " /"DECT " /"DECP "" "or
"ASSE ": stamp in its initial state (not factorized) "DECT"
: stamp entirely factorized "DEPT":
stamp partially factorized (possible only if LDLT) .REFA (9)

= "ms"/ "MR."" MS': symmetric
matrix "MR.": nonsymmetrical matrix
.REFA (10

) =/"NOEU" : the ddls of the matrix are carried by nodes/"GENE"
: the ddls of the matrix are generalized ddls. .REFA (11

) = "MPI_COMPLET "/"INCOMPLETE MPI_ "/"MATR _DISTR " "MPI_COMPLET
": Objects .VALM (and .CCVA) are "complete" "MPI_INCOMPLET
": Objects .VALM (and .CCVA) are "incomplete" because of a distributed computation MPI . Each processor assembles only the finite elements which are affected for him. "MATR_DISTR
": If MATR_DISTRIBUEE = ' OUI' for the MUMPS solver. This option dimensions with just the matrix assembled according to the number of unknowns present on the current processor. Note:

if "MPI _

INCOMPLETE", objects .VALM and .CCVA are allocated with their normal size, but they are partially filled from "0". if "MATR_DISTR", object .VALM is this time of a size smaller than in case "MPI_INCOMPLET ". DOCU (".REFA

") =/"ASSE" stamps initial,/"DECT"
stamps completely factorized,/"DECP"
stamps partially factorized. .VALM .VALMXD

3.2

V R/C NUM () This object

contains the values of the non-zero terms of the matrix (with the format Morse). There are 1 (or 2) element (S) in this collection (2 if the matrix is asymmetric). The 1st element

of the collection contains the upper part of the matrix, 2nd
contains the lower part. Note:

**** For***

*the nonsymmetrical matrixes, the diagonal is thus stored 2 times. One agrees to use only the diagonal stored in upper part (VALM (1)). * the arrangement of the terms of the matrix in the blocks is explained in the documentation of the sd _stockage [D4.06.07] .CONL .CONL*

3.3 S

V R dim =neq neq is

the number of equations of the system This optional

object is present only if there exists at least a d.o.f. of the type "LAGR": V (ieq) =
C if ieq corresponds to a DDL named "LAGR", 1. if not
. C is
the coefficient of conditioning of the ddls of Lagrange. .DIGS .DIGSS

3.4 V

R/C dim=2*neq neq is

the number of equations of the system This object

is present only if the matrix were factorized by "LDLT" or "MULT_FRONT " DIGS (1: neq

): values of the diagonal of initial matrix DIGS (neq
+1: 2*neq): values of the diagonal of factorized matrix .LIME Lists

3.5

names of the sd_matr_ elem having been assembled to give the sd_matr_ adze. This object exists only if the sd_matr_ adze comes from an assembly. It is
to better avoid making use of it. .UALF.

3.6 VALF, .WALF These collections

contain the terms of the factorized matrix. .UALF contains

factorized with LDLT (symmetric or NON-symmetric) .VALF contains
factorized with MULT_FRONT of part higher .WALF contains
factorized with MULT_FRONT of lower part .CCVA .CCVAS

3.7

V R/C object

.CCVA contains the columns of the initial matrix corresponding to the ddls to be eliminated (those which are imposed by a "kinematical" load). More precisely, one stores the submatrix of the terms on lines corresponding to DDLs free and columns with of DDLs imposed. .CCII .CCIIS

3.8

V I object

.CCII has same structure as object .CCVA. It V contains the indices of lines of the terms stored in .CCVA .CCLL

3.9 .CCLLS

I dim=3*nelim nelim is

the number of ddls "eliminated ". .CCLL ((I

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

the eliminated number of the equation corresponding to the ième d.o.f. , i1 is
the number of terms stored for the equation ieq i2 is
the cumulated number of terms stored for the equations J < I .CCID .CCIDS

3.10 V

I dim=neq+ 1 .CCID (ieq

) =1si the d.o.f. ieq eliminated , 0 if not
. CCID (neq
) = nelim: number of ddls eliminated .FETM .FETMS

3.11 V

K24 indirect (*) dim=nbsd (many subdomains) (*): sd

matr adze not FETI (i.e. FETM (K) .REFA (5) ("FETI" and for the time imposed on "MULT_FRONT "). JEVEUX object

listing the SD sd_matr_asse clean with each subdomain. FETM (I) = name

of the SD sd_matr_asse ième subdomain . .FETF .FETFS

3.12 V I

dim=nbsd optional

JEVEUX object indicating the character floating or not of a subdomain. .FETF (I) =

J (0<j<7) subdomain I comprises J modes of rigid bodies. non floating 0sous-field. .FETP .FETPXD

3.13 V

I LONG=nbsdf Lists indices

of “quasi-null” pivots of the floating subdomains. That is to say V=FETP

(I), V (J): jème

index of pivots of the ième floating subdomain. The LONMAX

of V (I) is equal to FETF (I). The number of

floating subdomains is accessible via attribute NUTIOC. The floating

subdomains are listed in the same order as the subdomains (floating or not) of the sd_feti. .FETR .FETRXD

3.14 V

R LONG=nbsdf Components

of the modes of rigid bodies. That is to say V=FETR

(I), V ((j-1) *nbddli

+k): kème component of the ième floating subdomain. nbddli,

the number of DDL of subdomain I, is stored in sd_feti.FETH (I). The number of

floating subdomains is accessible via attribute NUTIOC. The floating

subdomains are listed in the same order as the subdomains (floating or not) of the SD sd _feti. Complements

4 for FETI Data format

4.1 sd_matr_asse recursive In the case

of method FETI, data structure sd_matr_asse is recursive on two levels. A SD sd_matr_asse "main " , concerning the total field (REFA (5) =" FETI'), does not comprise any the usual JEVEUX objects , except the .REFA and the .LIME , with , on the other hand, besides the specific objects of the decomposition of fields: .FETR, FETF , .FETP and. FETM. This last is a pointer indicating the SD sd_matr_asse " slaves "associated with each local subdomains. These SD sd_matr_asse local are made up by the same JEVEUX objects that a sd _matr_asse (.VALE, .CONL …) mono-field usual. For time, the implementation of FETI in Code_Aster presupposes that these subdomains use all the same linear solver mono-field (REFA (5) = "MULT_FRONT" imposed by default, with thus a .VALF). This homogeneity facilitates handling of the local matrixes. sd_matr_asse

"main"------> sd_matr_asse "slave" (subdomain 1) (total field
) ------> sd 2) _matr_asse "slave" (subdomain object .FETM
… ------> sd
_matr_asse "slave FETI solver" (subdomain N) Rule
of

4.2 naming In the case

of one, one arbitrarily chose (in assmam.f) the following rule of naming for the SD sd_matr_asse slaves dependant each on a subdomain: nom_de_la_SD_sd _matr_asse_maître (1:11) "F" chaîne_de_caractères_libre (2:8) the free character string is generated by a call to the routine gcncon.f. Typical case of

4.2.1 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 the data structure sd_feti [D4.06.21]). The SD sd_matr_asse "main " is always built, but its pointer .FETM will indicate only the subdomains concerned with the processor running: .FETM (jk) will be one K24 valid that if the jk subdomain is in the perimeter of the processor J. The same applies to objects .FETF, .FETP and .FETR which are filled only if necessary. For the processor

J: sd_matr_asse "main

" ------> "vacuum" (total field)
------> sd_matr_ adze "slave " (j1 subdomain) object .FETM ------
> sd_matr_ adze "slave " (j2 subdomain) ------>…
the j1

subdomains, j2,… which is those are concerned with the processor J